FIG. 1

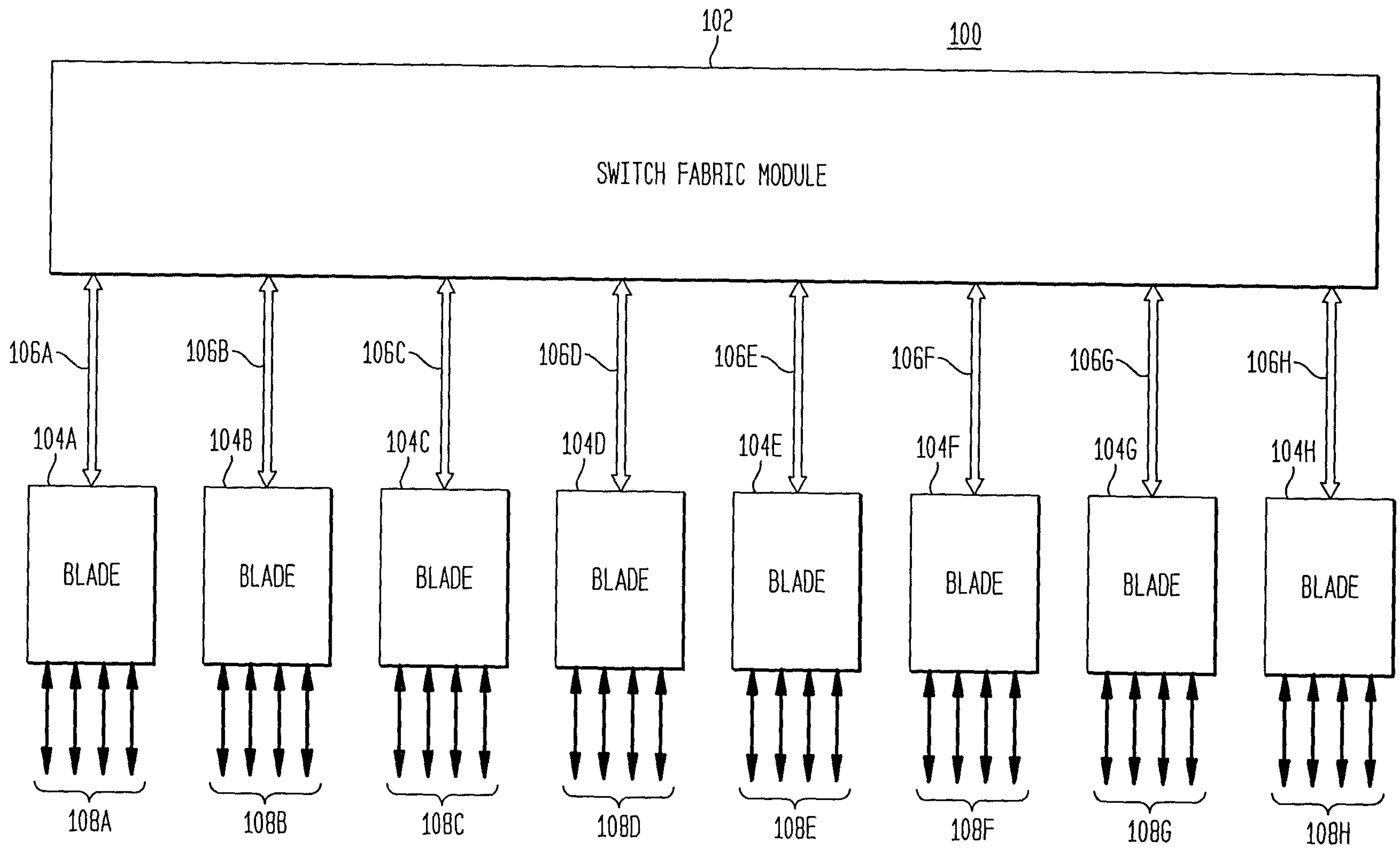

102
100
SWITCH FABRIC MODULE
106A
106B
106C
106D
106E
106F
106G
106H
104A
104B
104C
104D
104E
104F
104G
104H
BLADE
BLADE
BLADE
BLADE
BLADE
BLADE
BLADE
BLADE
108A
108B
108C
108D
108E
108F
108G
108H

FIG. 2

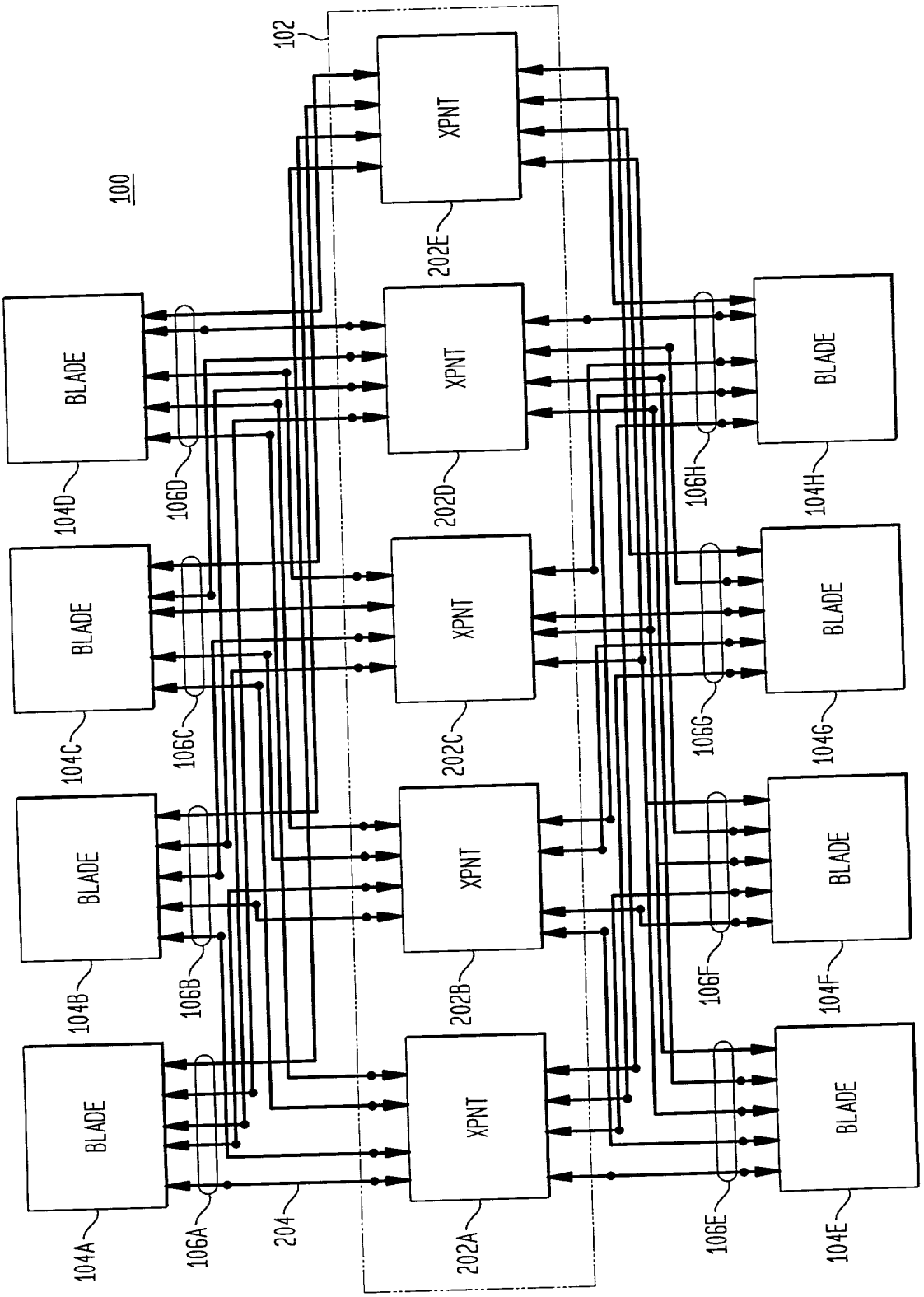

100
102
XPNT
202A
202B
202C
202D
202E
BLADE
104A
104B
104C
104D
104E
104F
104G
104H
106A
106B
106C
106D
106E
106F
106G
106H
204

FIG. 3A

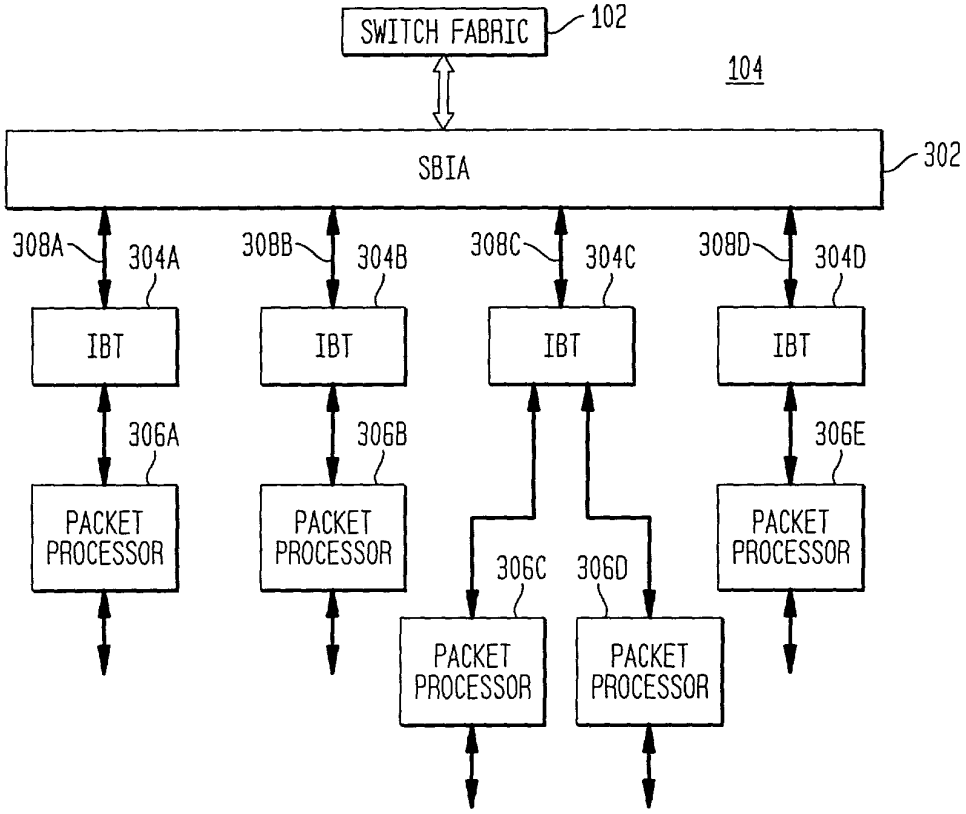

SWITCH FABRIC
102
104
SBIA
302
308A
304A
308B
304B
308C
304C
308D
304D
IBT
IBT
IBT
IBT
306A
306B
306E
PACKET PROCESSOR
PACKET PROCESSOR
PACKET PROCESSOR
306C
306D
PACKET PROCESSOR
PACKET PROCESSOR

FIG. 3B

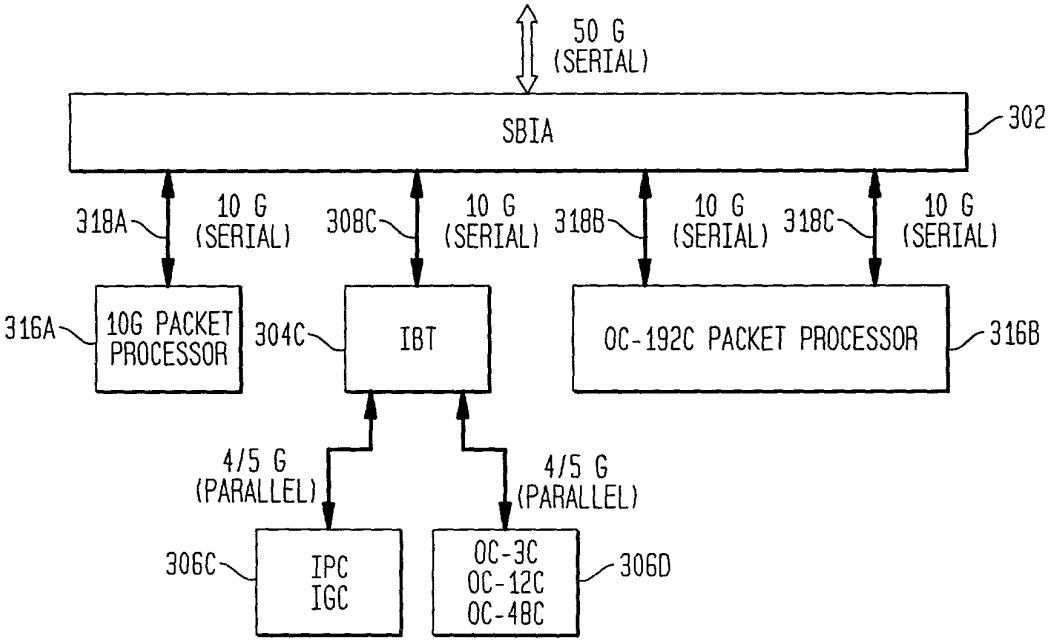

50 G (SERIAL)
SBIA
302
318A
10 G (SERIAL)
308C
10 G (SERIAL)
318B
10 G (SERIAL)
318C
10 G (SERIAL)
316A
10G PACKET PROCESSOR
304C
IBT
OC-192C PACKET PROCESSOR
316B
4/5 G (PARALLEL)
4/5 G (PARALLEL)
306C
IPC IGC
OC-3C OC-12C OC-48C
306D

FIG. 4

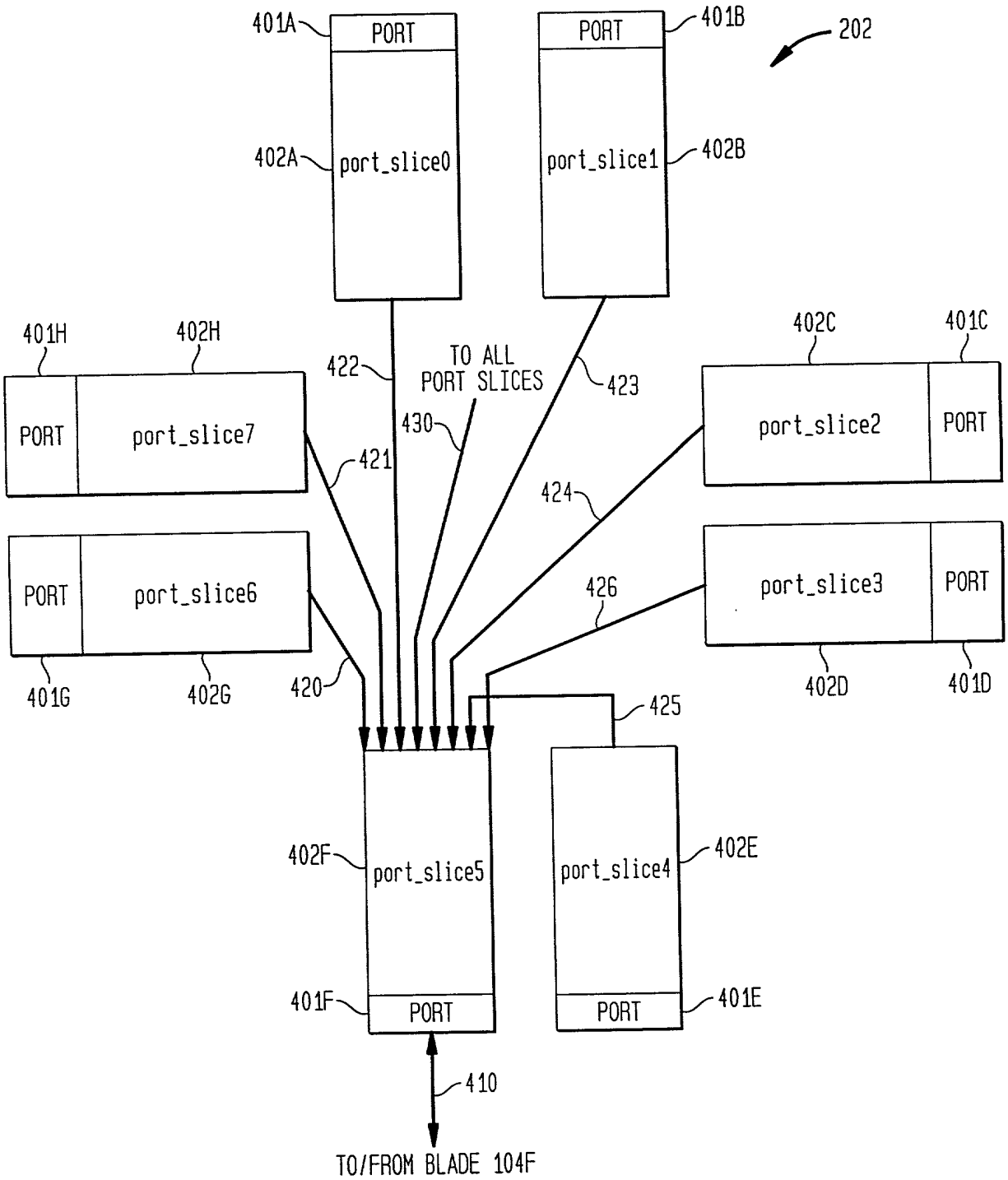

401A
PORT
PORT
401B
202
402A
port_slice0
port_slice1
402B
401H
402H
422
TO ALL
PORT SLICES
423
402C
401C
PORT
port_slice7
430
port_slice2
PORT
421
424
PORT
port_slice6
426
port_slice3
PORT
401G
402G
420
425
402D
401D
402F
port_slice5
port_slice4
402E
401F
PORT
PORT
401E
410
TO/FROM BLADE 104F

FIG. 5

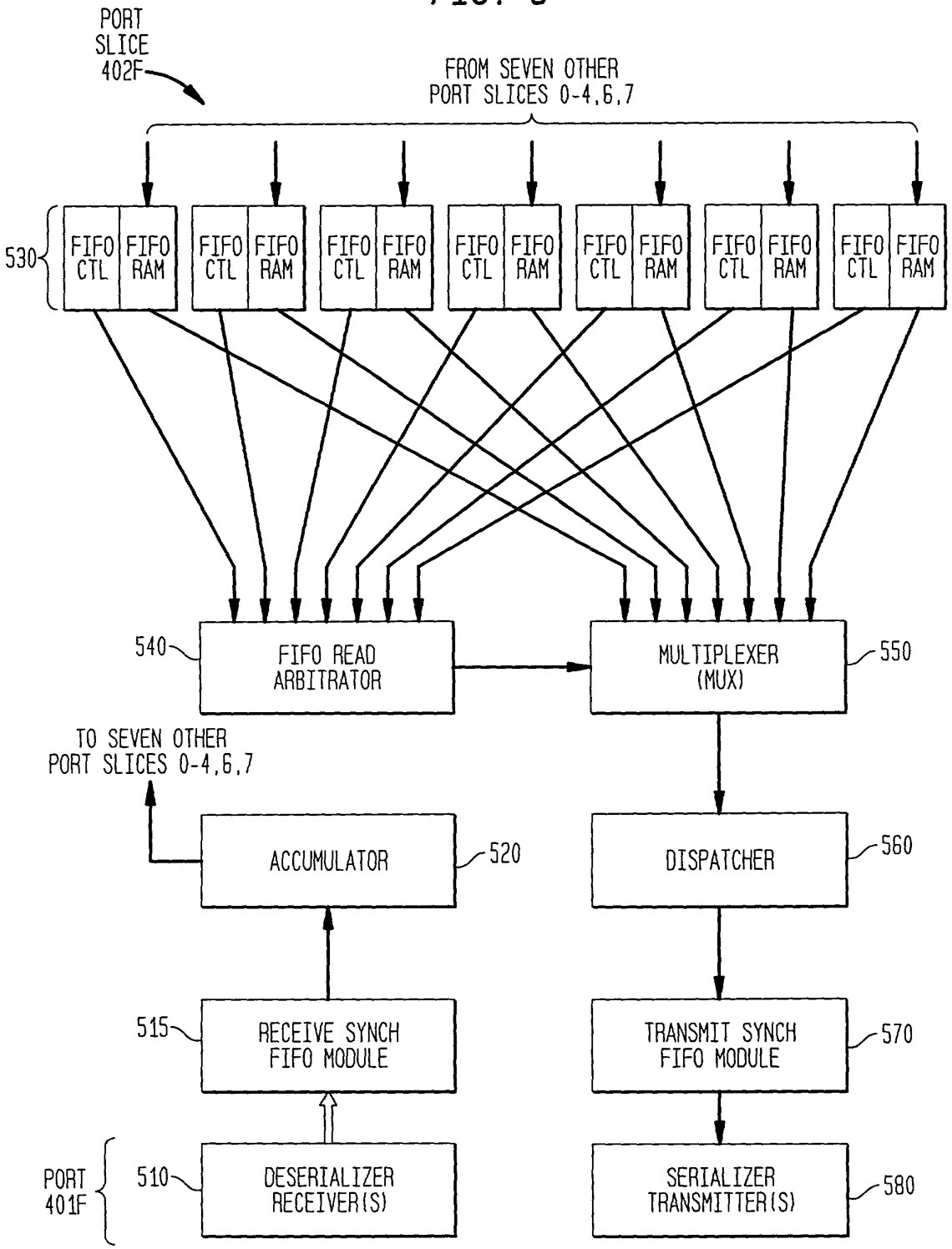

PORT SLICE 402F
FROM SEVEN OTHER PORT SLICES 0-4,6,7
530
FIFO CTL
FIFO RAM
540
FIFO READ ARBITRATOR
MULTIPLEXER (MUX)
550
TO SEVEN OTHER PORT SLICES 0-4,6,7
ACCUMULATOR
520
DISPATCHER
560
515
RECEIVE SYNCH FIFO MODULE
TRANSMIT SYNCH FIFO MODULE
570
PORT 401F
510
DESERIALIZER RECEIVER(S)
SERIALIZER TRANSMITTER(S)
580

FIG. 6

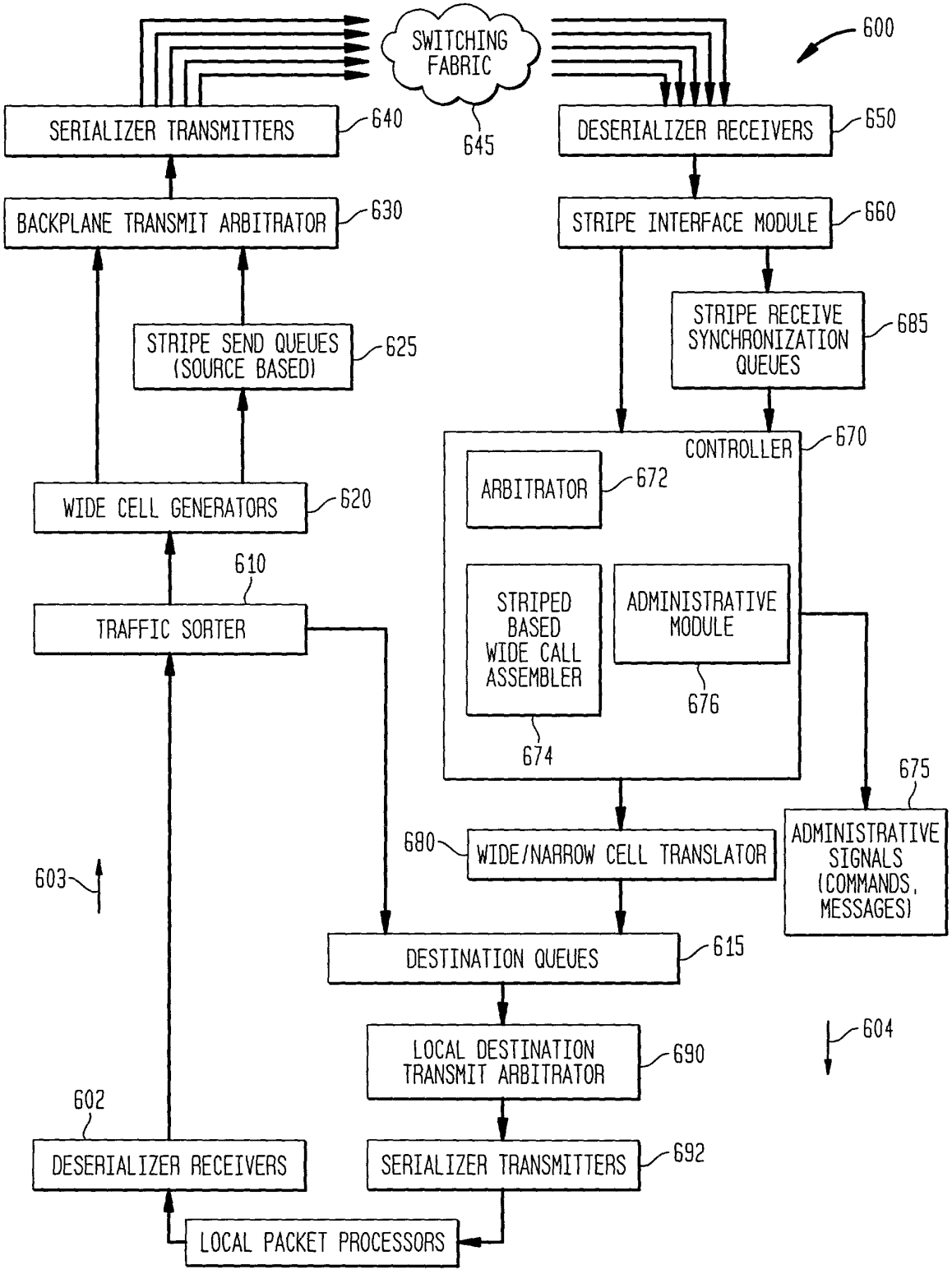

SWITCHING FABRIC
600
SERIALIZER TRANSMITTERS
640
645
DESERIALIZER RECEIVERS
650
BACKPLANE TRANSMIT ARBITRATOR
630
STRIPE INTERFACE MODULE
660
STRIPE SEND QUEUES (SOURCE BASED)
625
STRIPE RECEIVE SYNCHRONIZATION QUEUES
685
CONTROLLER
670
ARBITRATOR
672
WIDE CELL GENERATORS
620
610
STRIPED BASED WIDE CALL ASSEMBLER
ADMINISTRATIVE MODULE
676
TRAFFIC SORTER
674
675
ADMINISTRATIVE SIGNALS (COMMANDS, MESSAGES)
680
WIDE/NARROW CELL TRANSLATOR
603
DESTINATION QUEUES
615
LOCAL DESTINATION TRANSMIT ARBITRATOR
690
604
602
DESERIALIZER RECEIVERS
SERIALIZER TRANSMITTERS
692
LOCAL PACKET PROCESSORS

FIG. 7

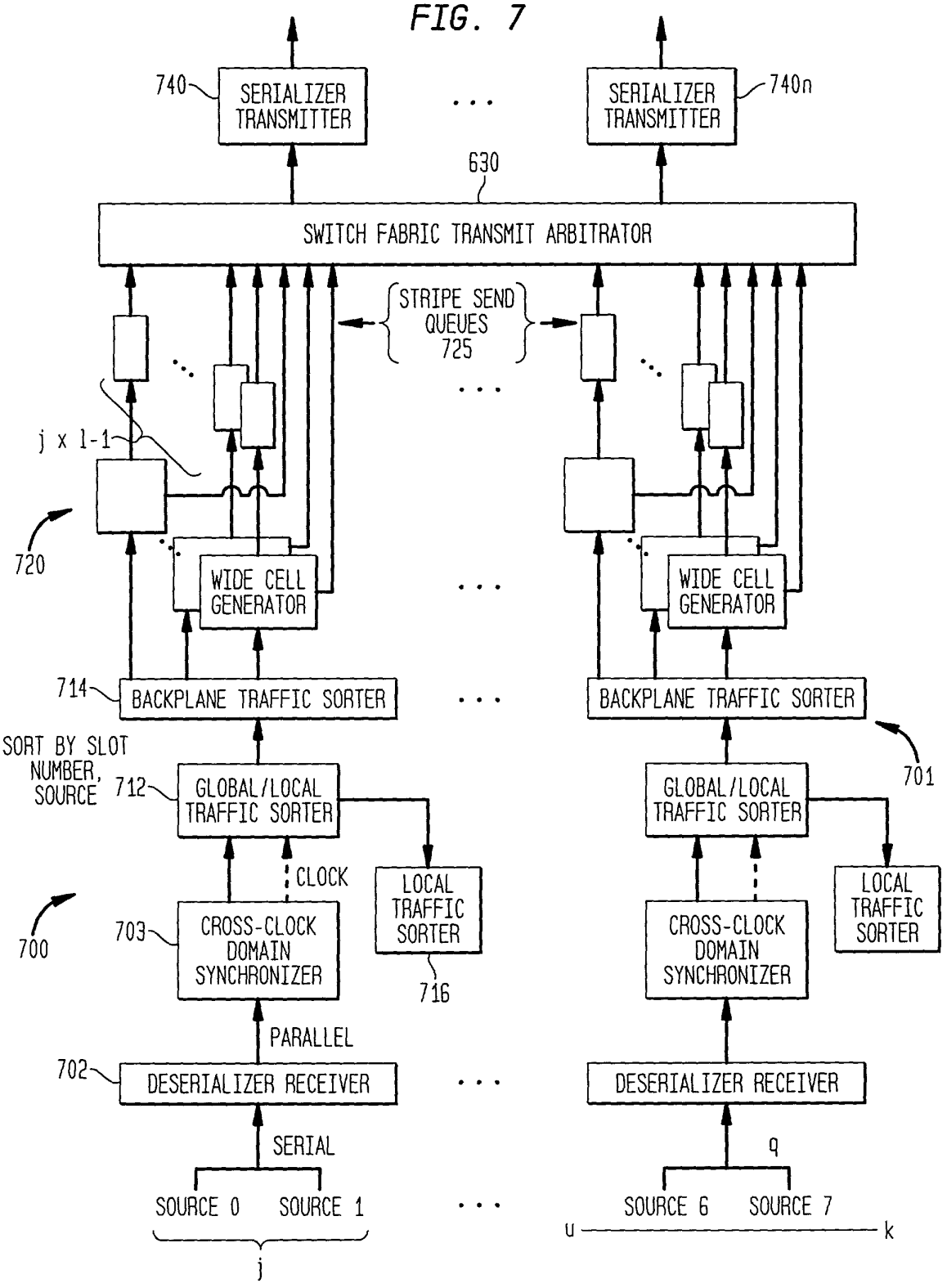

740
SERIALIZER TRANSMITTER
SERIALIZER TRANSMITTER
740n
630
SWITCH FABRIC TRANSMIT ARBITRATOR
STRIPE SEND QUEUES 725
j x 1-1
720
WIDE CELL GENERATOR
WIDE CELL GENERATOR
714
BACKPLANE TRAFFIC SORTER
BACKPLANE TRAFFIC SORTER
701
SORT BY SLOT NUMBER, SOURCE
712
GLOBAL/LOCAL TRAFFIC SORTER
GLOBAL/LOCAL TRAFFIC SORTER
CLOCK
700
703
CROSS-CLOCK DOMAIN SYNCHRONIZER
CROSS-CLOCK DOMAIN SYNCHRONIZER
LOCAL TRAFFIC SORTER
LOCAL TRAFFIC SORTER
716
PARALLEL
702
DESERIALIZER RECEIVER
DESERIALIZER RECEIVER
SERIAL
q
SOURCE 0
SOURCE 1
SOURCE 6
SOURCE 7
j
u
k

FIG. 8

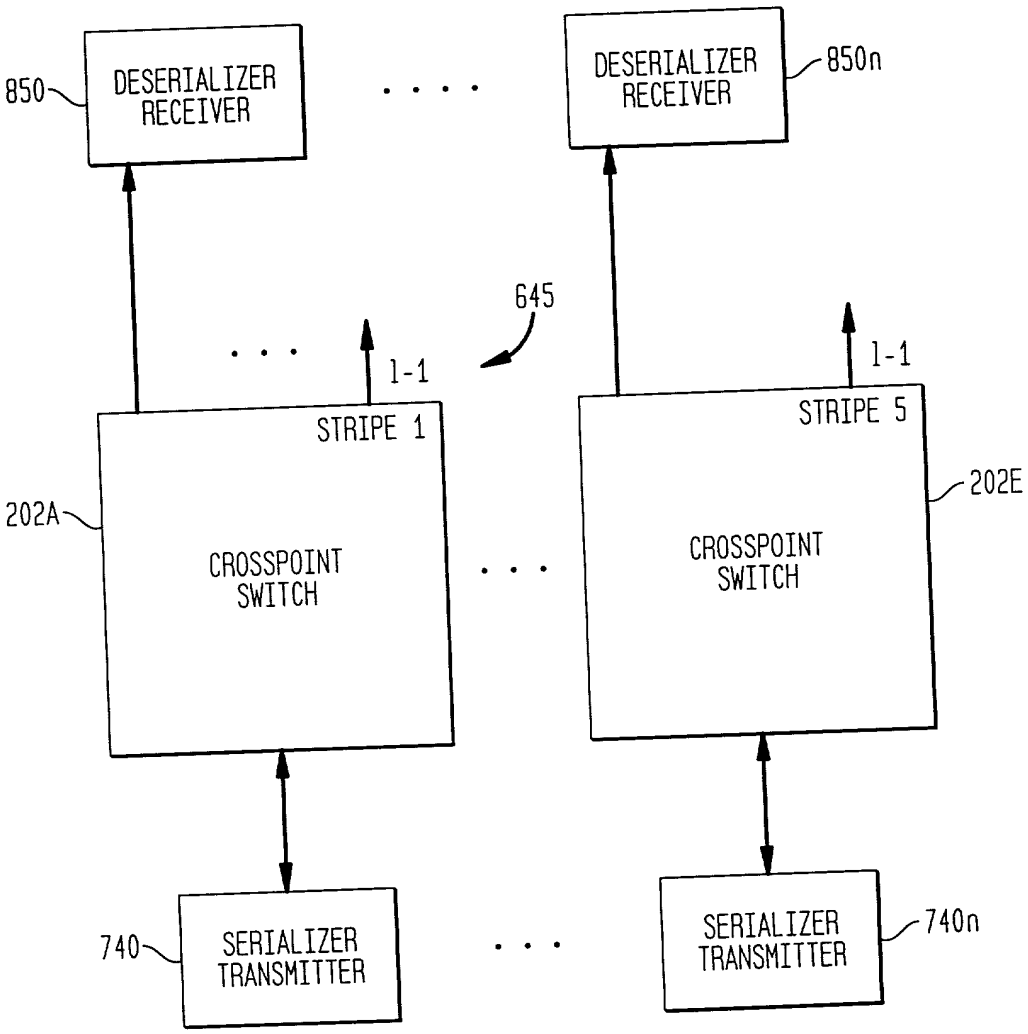

850
DESERIALIZER RECEIVER
DESERIALIZER RECEIVER
850n
645
1-1
1-1
STRIPE 1
STRIPE 5
202A
202E
CROSSPOINT SWITCH
CROSSPOINT SWITCH
740
SERIALIZER TRANSMITTER
SERIALIZER TRANSMITTER
740n

FIG. 9

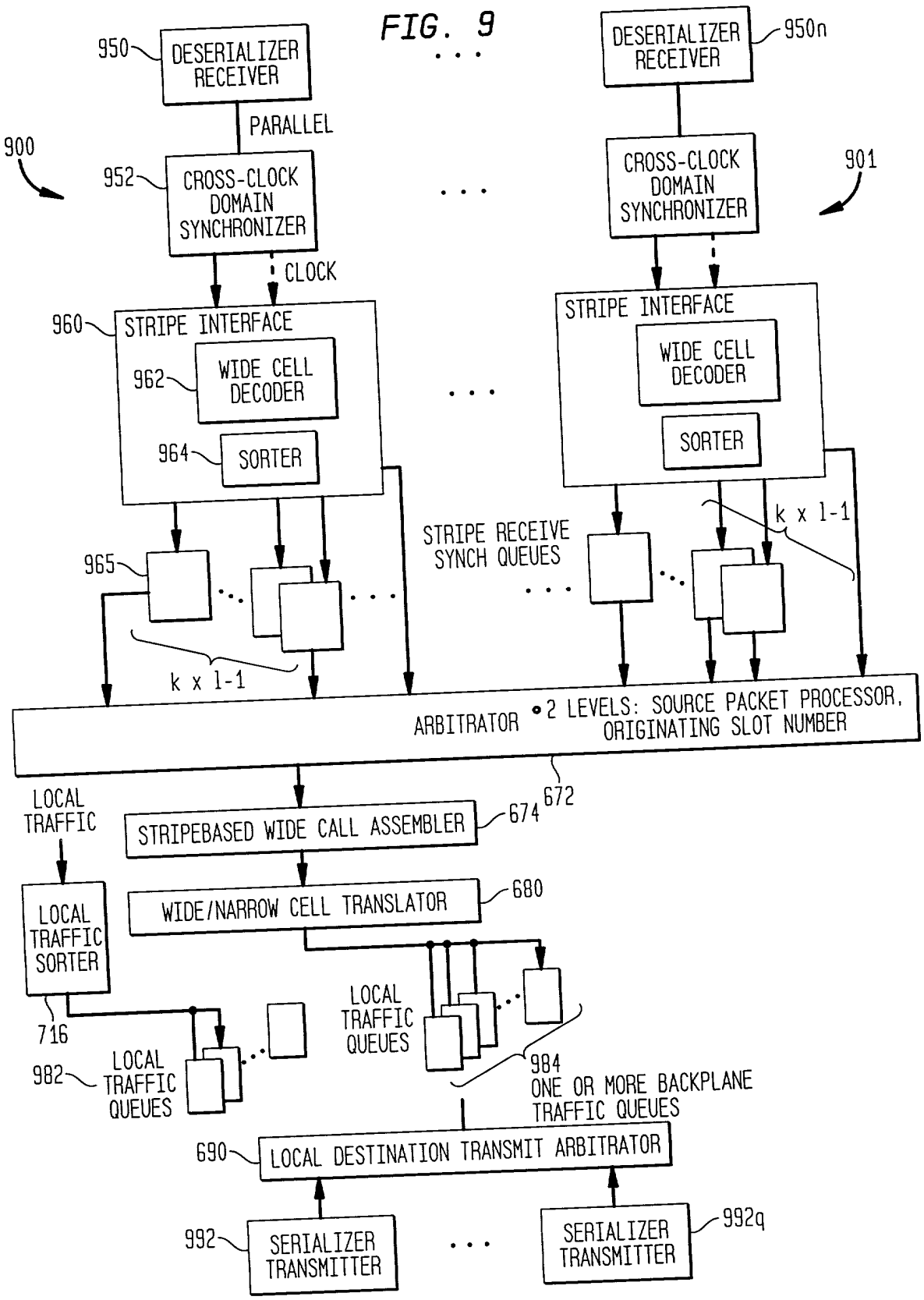

950
DESERIALIZER RECEIVER
950n
DESERIALIZER RECEIVER
PARALLEL
900
952
CROSS-CLOCK DOMAIN SYNCHRONIZER
901
CROSS-CLOCK DOMAIN SYNCHRONIZER
CLOCK
960
STRIPE INTERFACE
962
WIDE CELL DECODER
964
SORTER
STRIPE INTERFACE
WIDE CELL DECODER
SORTER
965
STRIPE RECEIVE SYNCH QUEUES
k x 1-1
k x 1-1
ARBITRATOR
2 LEVELS: SOURCE PACKET PROCESSOR, ORIGINATING SLOT NUMBER
672
LOCAL TRAFFIC
STRIPEBASED WIDE CALL ASSEMBLER
674
LOCAL TRAFFIC SORTER
WIDE/NARROW CELL TRANSLATOR
680
716
LOCAL TRAFFIC QUEUES
982
LOCAL TRAFFIC QUEUES
984
ONE OR MORE BACKPLANE TRAFFIC QUEUES
690
LOCAL DESTINATION TRANSMIT ARBITRATOR
992
SERIALIZER TRANSMITTER
SERIALIZER TRANSMITTER
992q

FIG. 10

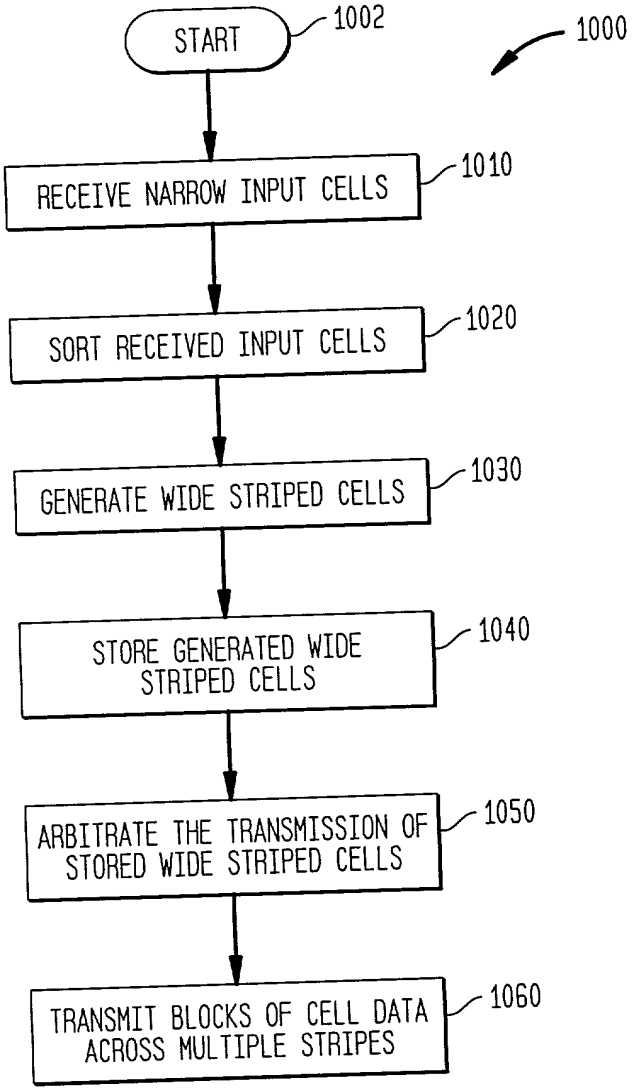

START
1002
1000
RECEIVE NARROW INPUT CELLS
1010
SORT RECEIVED INPUT CELLS
1020
GENERATE WIDE STRIPED CELLS
1030
STORE GENERATED WIDE STRIPED CELLS
1040
ARBITRATE THE TRANSMISSION OF STORED WIDE STRIPED CELLS
1050
TRANSMIT BLOCKS OF CELL DATA ACROSS MULTIPLE STRIPES
1060

FIG. 11

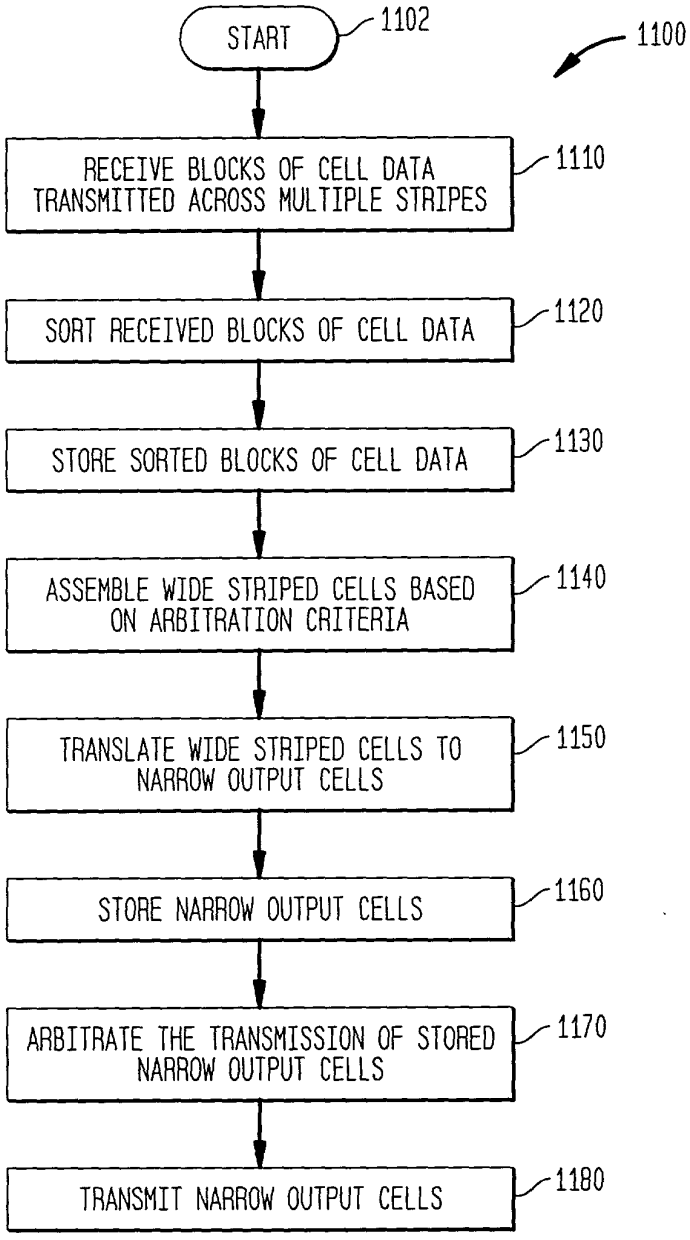

START
1102
1100
RECEIVE BLOCKS OF CELL DATA TRANSMITTED ACROSS MULTIPLE STRIPES
1110
SORT RECEIVED BLOCKS OF CELL DATA
1120
STORE SORTED BLOCKS OF CELL DATA
1130
ASSEMBLE WIDE STRIPED CELLS BASED ON ARBITRATION CRITERIA
1140
TRANSLATE WIDE STRIPED CELLS TO NARROW OUTPUT CELLS
1150
STORE NARROW OUTPUT CELLS
1160
ARBITRATE THE TRANSMISSION OF STORED NARROW OUTPUT CELLS
1170
TRANSMIT NARROW OUTPUT CELLS
1180

FIG. 12

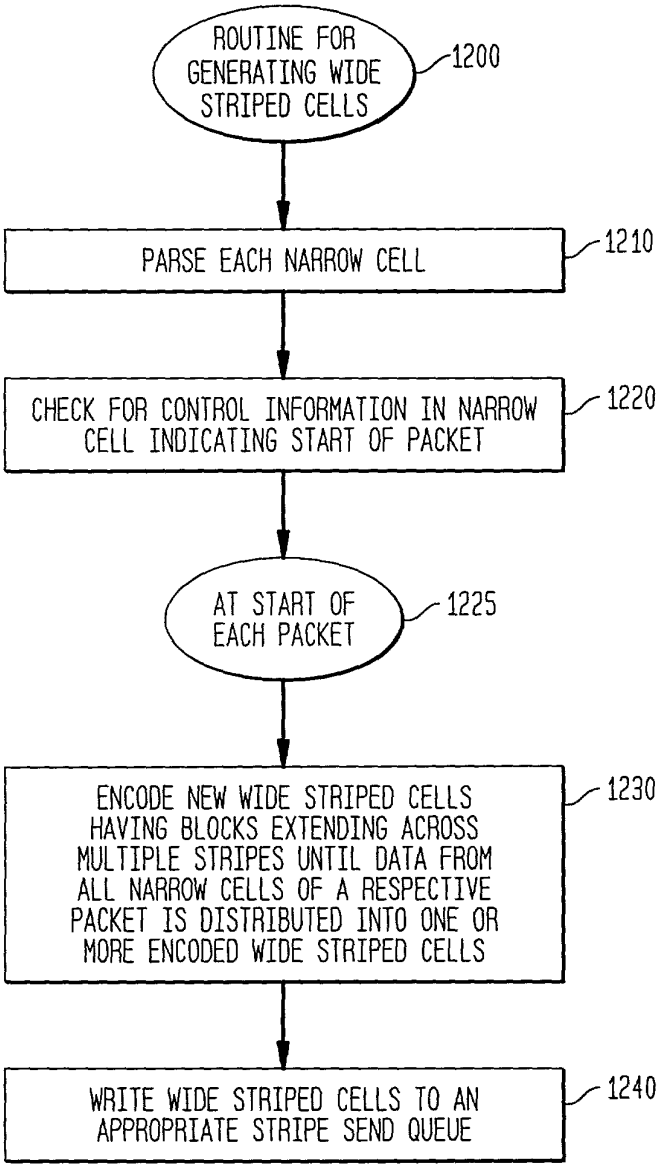

ROUTINE FOR GENERATING WIDE STRIPED CELLS
1200
PARSE EACH NARROW CELL
1210
CHECK FOR CONTROL INFORMATION IN NARROW CELL INDICATING START OF PACKET
1220
AT START OF EACH PACKET
1225
ENCODE NEW WIDE STRIPED CELLS HAVING BLOCKS EXTENDING ACROSS MULTIPLE STRIPES UNTIL DATA FROM ALL NARROW CELLS OF A RESPECTIVE PACKET IS DISTRIBUTED INTO ONE OR MORE ENCODED WIDE STRIPED CELLS
1230
WRITE WIDE STRIPED CELLS TO AN APPROPRIATE STRIPE SEND QUEUE
1240

FIG. 13

1300

LANE 0	LANE 1	LANE 2	LANE 3
CONTROL INFORMATION	STATE INFORMATION	RESERVED	RESERVED
D0	D1	D2	D3
D4	D5	D6	D7
D8	D9	D10	D11
D12	D13	D14	D15
⋮	⋮	⋮	⋮
D28	D29	D30	D31

1310

STATE INFORMATION	
NAME	DESCRIPTION
SLOT NUMBER	DESTINATION SLOT NUMBER WHERE CELL DATA BEING SENT
PAYLOAD STATE	RESERVED, SOP, DATA, ABORT
SOURCE OR DESTINATION PACKET PROCESSOR IDENTIFIER	ENCODED NUMBER IDENTIFYING A SOURCE OR DESTINATION PACKET PROCESSOR
RESERVED	RESERVED

FIG. 14

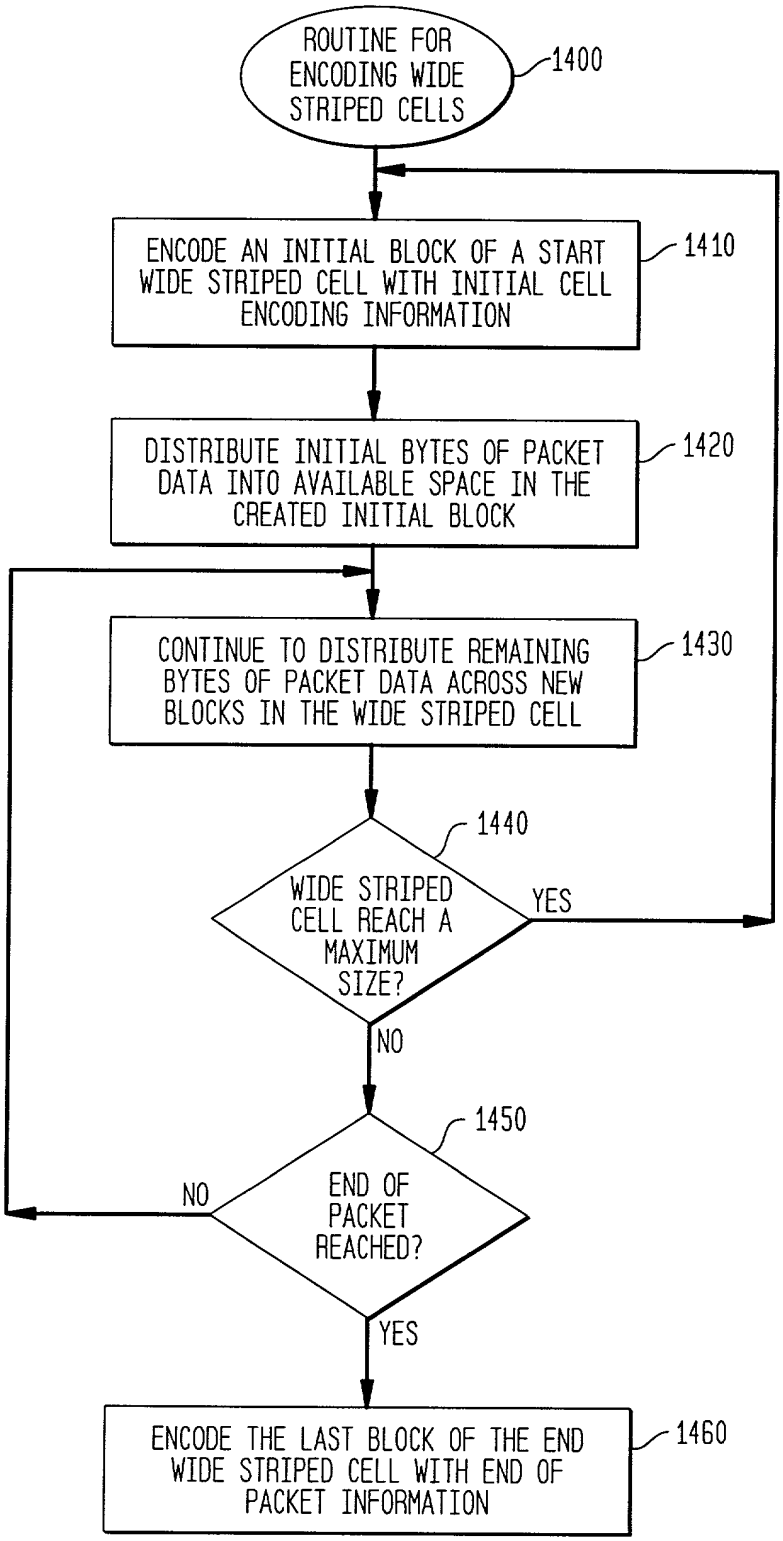

ROUTINE FOR ENCODING WIDE STRIPED CELLS
1400
ENCODE AN INITIAL BLOCK OF A START WIDE STRIPED CELL WITH INITIAL CELL ENCODING INFORMATION
1410
DISTRIBUTE INITIAL BYTES OF PACKET DATA INTO AVAILABLE SPACE IN THE CREATED INITIAL BLOCK
1420
CONTINUE TO DISTRIBUTE REMAINING BYTES OF PACKET DATA ACROSS NEW BLOCKS IN THE WIDE STRIPED CELL
1430
1440
WIDE STRIPED CELL REACH A MAXIMUM SIZE?
YES
NO
1450
END OF PACKET REACHED?
NO
YES
ENCODE THE LAST BLOCK OF THE END WIDE STRIPED CELL WITH END OF PACKET INFORMATION
1460

FIG. 15A

	STRIPE 1				STRIPE 2				STRIPE 3				STRIPE 4				STRIPE 5			
CYCLE	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3
1	K0	STATE	D0	D1	K0	STATE	D2	D3	K0	STATE	D4	D5	K0	STATE	D6	D7	K0	STATE	RES	RES
2	D8																			D27
3	D28																			D47
4	D48																			D67
5	D68																			D87
6	D88																			D107
7	D108																			D127
8	D128																			D147

1500

FIG. 15B

STATE INFORMATION	
NAME	DESCRIPTION
SLOT NUMBER	DESTINATION SLOT NUMBER FOR BIA TO CROSSPOINT SWITCH DIRECTION SOURCE SLOT NUMBER FOR CROSSPOINT SWITCH TO BIA DIRECTION
PAYLOAD STATE	ENCODED PAYLOAD STATE INFORMATION (RESERVED, SOA, DATA, ABORT)
RESERVED	RESERVED

FIG. 15C

END OF PACKET ENCODING INFORMATION

1. EOP DURING CYCLE 1 (ie. DURING TRANSMISSION OF STATE INFORMATION)

1	K0	state	D0	D1	K0	state	D2	D3	K0	state	K1	K1	K0	state	K1	K1	K0	state	RES	RES

NOTE THAT THE K0, STATE, AND RESERVED BYTES ARE ALL PRESERVED, AS IN ANY OTHER CYCLE 1 TRANMISSION. THE K1 CHARACTER IS TREATED AS DATA

2. EOP DURING CYCLE n (n!=0)

1	K0	state	D0	D1	K0	state	D2	D3	K0	state	D4	D5	K0	state	D6	D7	K0	state	RES	RES
2	D8																			D27
3	D28				D32	D33	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1

3. EOP AT BLOCK BOUNDARY DURING CYCLE n (n!=8)

1	K0	state	D0	D1	K0	state	D2	D3	K0	state	D4	D5	K0	state	D6	D7	K0	state	RES	RES
2	D8																			D27
3	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1	K1

NOTE THAT WHEN n>0, THE BLOCK BOUNDARY FOR DATA IS IN LANE 3 STRIPE 5. HOWEVER, FOR n=0. THE BLOCK BOUNDARY FOR DATA IS IN LANE 3 OF STRIPE 4.

4. EOP at cell boundary

6	D88																		D107
7	D108																		D127
8	D128																		D147

1	K0	state	K1	K1	K0	state	K1	K1	K0	state	K1	K1	K0	state	K1	K1	K0	state	RES	RES

FIG. 15D

	STRIPE 1				STRIPE 2				STRIPE 3				STRIPE 4				STRIPE 5			
CYCLE	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3
1	K0	P1	D0	D1																
2	D8			D11																
3	D28			D31	K0	P1	D2	D3												
4	D48			D51	D12			D15					K0	P1	D6	D7				
5	D68			D71	D32			D35					D20			D23				
6	D88			D91	D52			D55	K0	P1	D4	D5	D40			D43				
7	D108			D111	D72			D75	D16			D19	D60			D63	K0	P1	RES	RES
8	D128			D131	D92			D95	D36			D39	D80			D83	D24			D27

FIG. 16

	STRIPE 1				STRIPE 2				STRIPE 3				STRIPE 4				STRIPE 5			
CYCLE	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3
1	K0	SS1	D0	D1	K0	SS6	D151	D152	K0	SS2	D4	D5	K0	SS3	D6	D7	K0	SS7	RES	RES
2	D8			D11	D161	D162	D163	K1	D16			D19	D20			D23	D320			D323
3	D28			D31	K0	SS2	D2	D3	D36			D39	D40			D43	D340			D343
4	D48			D51	D12			D15	D56			D59	K0	SS4	D6	D7	D360			D363
5	D68			D71	D32			D35	D76	K1	K1	K1	D20			D23	D380			D383
6	D88			D91	D52			D55	K0	SS3	D4	D5	D40			D43	K1	K1	K1	K1
7	D108			D111	D72			D75	D16			D19	D60			D63	K0	SS5	RES	RES
8	D128			D131	K0	SS5	D2	D3	D36			D39	D80	K1	K1	K1	D24			D27
9	K0	SS4	D0	D1	D12			D15	K0	SS6	D153	D154	K0	SS2	D6	D7	D44			D47
10	D8			D11	D32			D35	K1	K1	K1	K1	D20			D23	D64			D67
11	D28			D31	D52			D55	K0	SS7	D300	D301	D40			D43	K0	SS1	RES	RES
12	D48			D51	K0	SS1	D2	D3	D312			D315	D60			D63	D24			D27
13	D68			D71	D12			D15	D332			D335	K1	K1	K1	K1	D44			D47
14	K0	SS7	D296	D297	D32			D35	D352			D355	K0	SS6	D155	D156	D64			D67
15	D304			D307	D52			D55	D372			D375	K1	K1	K1	K1	D84			D87
16	D324			D327	D72			D75	K1	K1	K1	K1	K0	SS1	D6	D7	D104			D107
17	D344			D347	D92			D95	K0	SS5	D4	D5	D20			D23	D124			D127
18	D364			D367	D112			D115	D16			D19	D40			D43	D144			D147
19	K1	K1	K1	K1	D132			D135	D36			D39	D60			D63	D0	SS3	RES	RES
20	K0	SS6	D149	D150	K0	SS7	D298	D299	D56			D59	D80			D83	D24			D27
21	D157			D160	D308			D311	K0	SS1	D4	D5	D100			D103	D44	D45	K1	K1
22	K0	SS1	K1	K1	D328			D331	D16			D19	D120			D123	K0	SS2	RES	RES
23	K0	SS3	D0	D1	D348			D351	D36			D39	D140			D143	D24			D27
24	D8			D11	D368			D371	D56			D59	K0	SS7	D302	D303	D44			D47

GREEN YELLOW ORANGE BLUE RED RUST PINK

FIG. 17

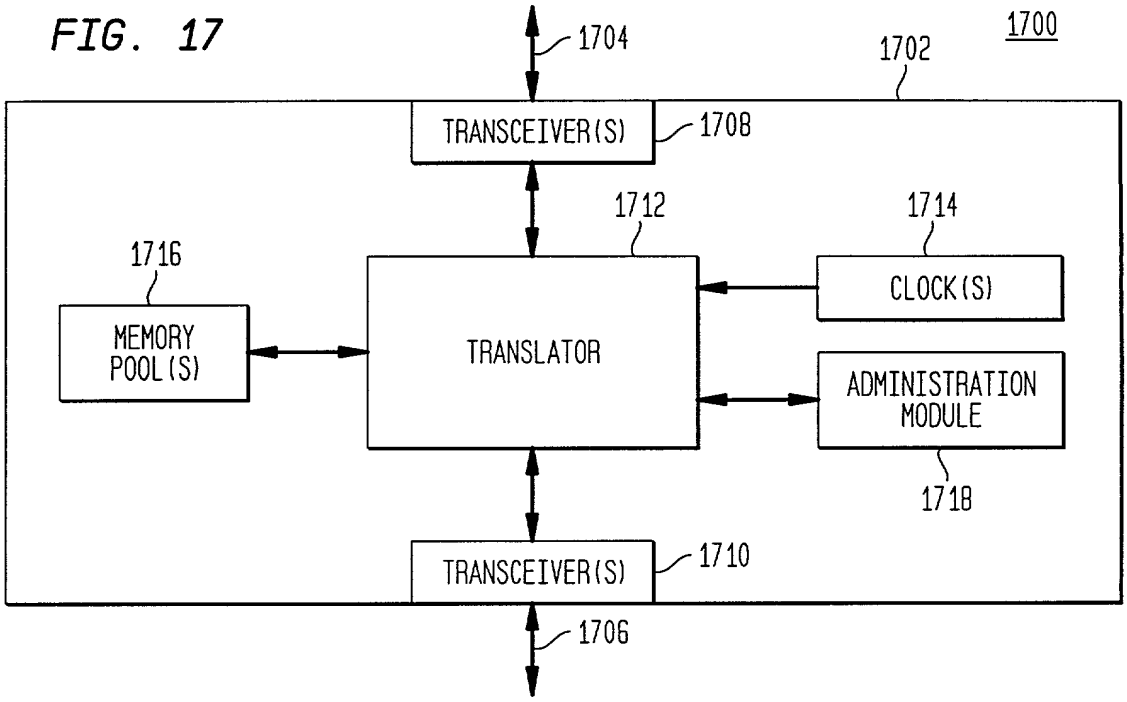

1700
1704
1702
TRANSCEIVER(S)
1708
1712
1714
1716
CLOCK(S)
MEMORY POOL(S)
TRANSLATOR
ADMINISTRATION MODULE
1718
TRANSCEIVER(S)
1710
1706

FIG. 18

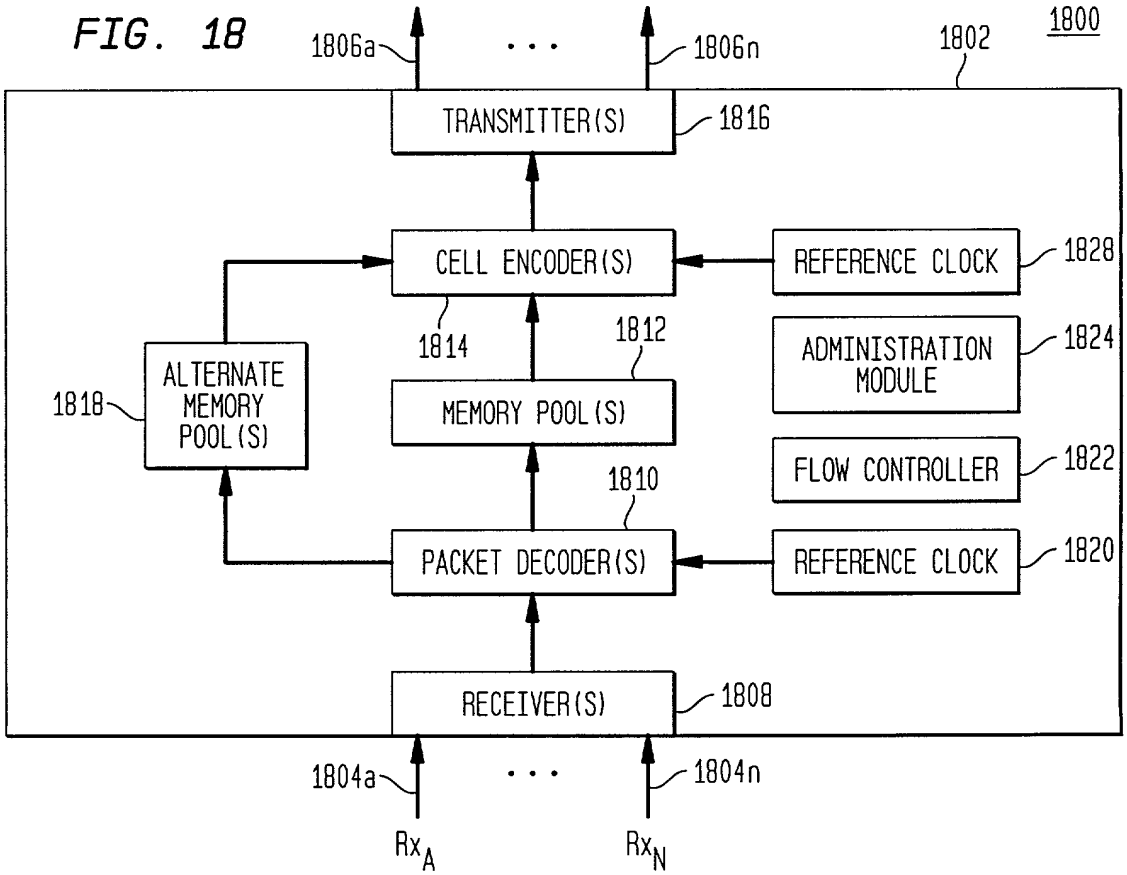

1800
1806a
1806n
1802
TRANSMITTER(S)
1816
CELL ENCODER(S)
REFERENCE CLOCK
1828
1814
1812
ADMINISTRATION MODULE
1824
1818
ALTERNATE MEMORY POOL(S)
MEMORY POOL(S)
FLOW CONTROLLER
1822
1810
PACKET DECODER(S)
REFERENCE CLOCK
1820
RECEIVER(S)
1808
1804a
1804n
Rx_A
Rx_N

FIG. 19

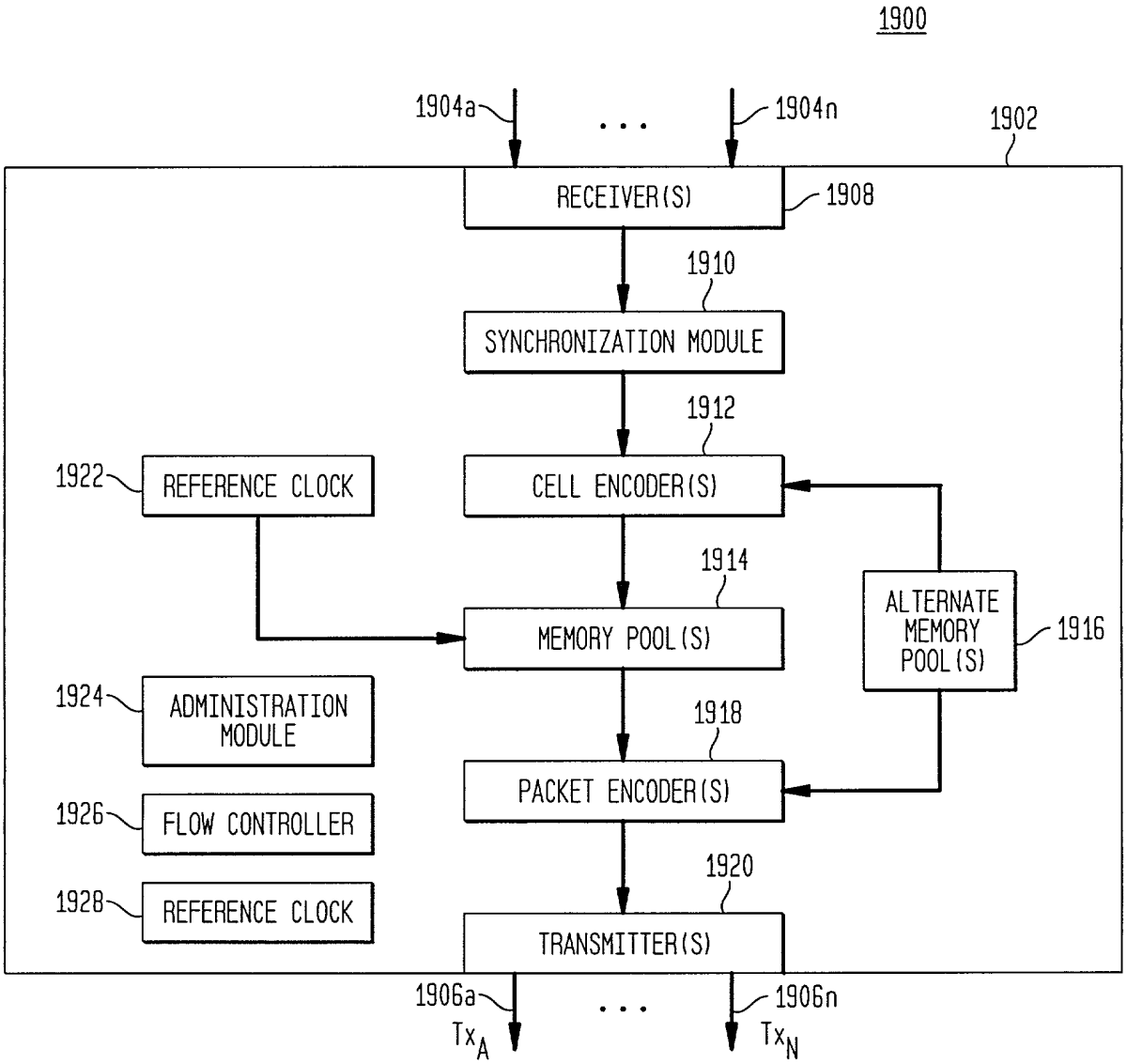

1900
1904a
1904n
1902
RECEIVER(S)
1908
1910
SYNCHRONIZATION MODULE
1912
1922
REFERENCE CLOCK
CELL ENCODER(S)
1914
ALTERNATE MEMORY POOL(S)
1916
MEMORY POOL(S)
1924
ADMINISTRATION MODULE
1918
PACKET ENCODER(S)
1926
FLOW CONTROLLER
1920
1928
REFERENCE CLOCK
TRANSMITTER(S)
1906a
1906n
Tx_A
Tx_N

FIG. 20

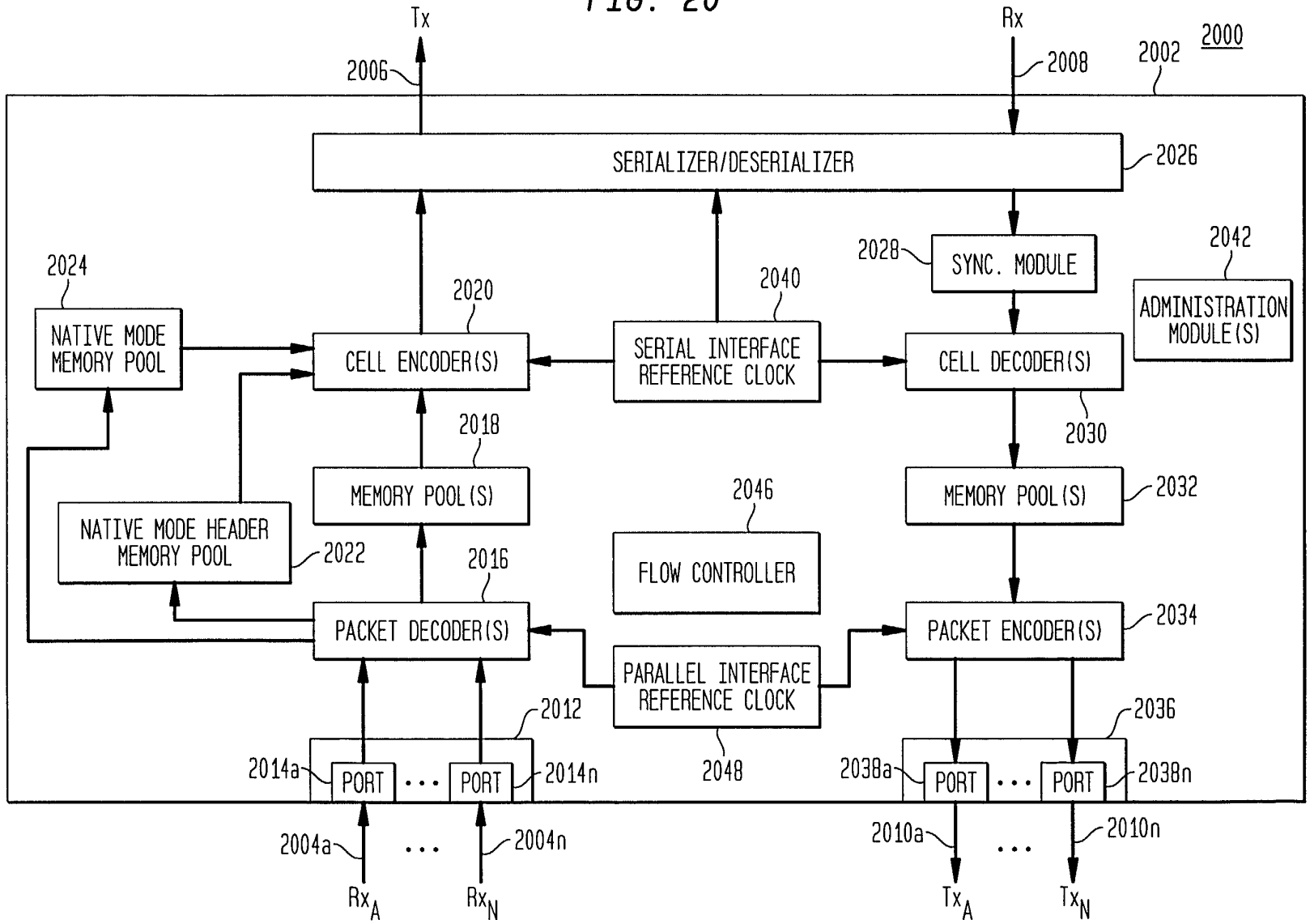

Tx
Rx
2000
2002
2006
2008
SERIALIZER/DESERIALIZER
2026
2028
SYNC. MODULE
2042
ADMINISTRATION MODULE(S)
2024
NATIVE MODE MEMORY POOL
2020
CELL ENCODER(S)
2040
SERIAL INTERFACE REFERENCE CLOCK
CELL DECODER(S)
2030
2018
MEMORY POOL(S)
2046
MEMORY POOL(S)
2032
NATIVE MODE HEADER MEMORY POOL
2022
2016
FLOW CONTROLLER
PACKET DECODER(S)
PACKET ENCODER(S)
2034
PARALLEL INTERFACE REFERENCE CLOCK
2012
2036
2014a
PORT
PORT
2014n
2048
2038a
PORT
PORT
2038n
2004a
2004n
2010a
2010n
Rx_A
Rx_N
Tx_A
Tx_N

FIG. 21A

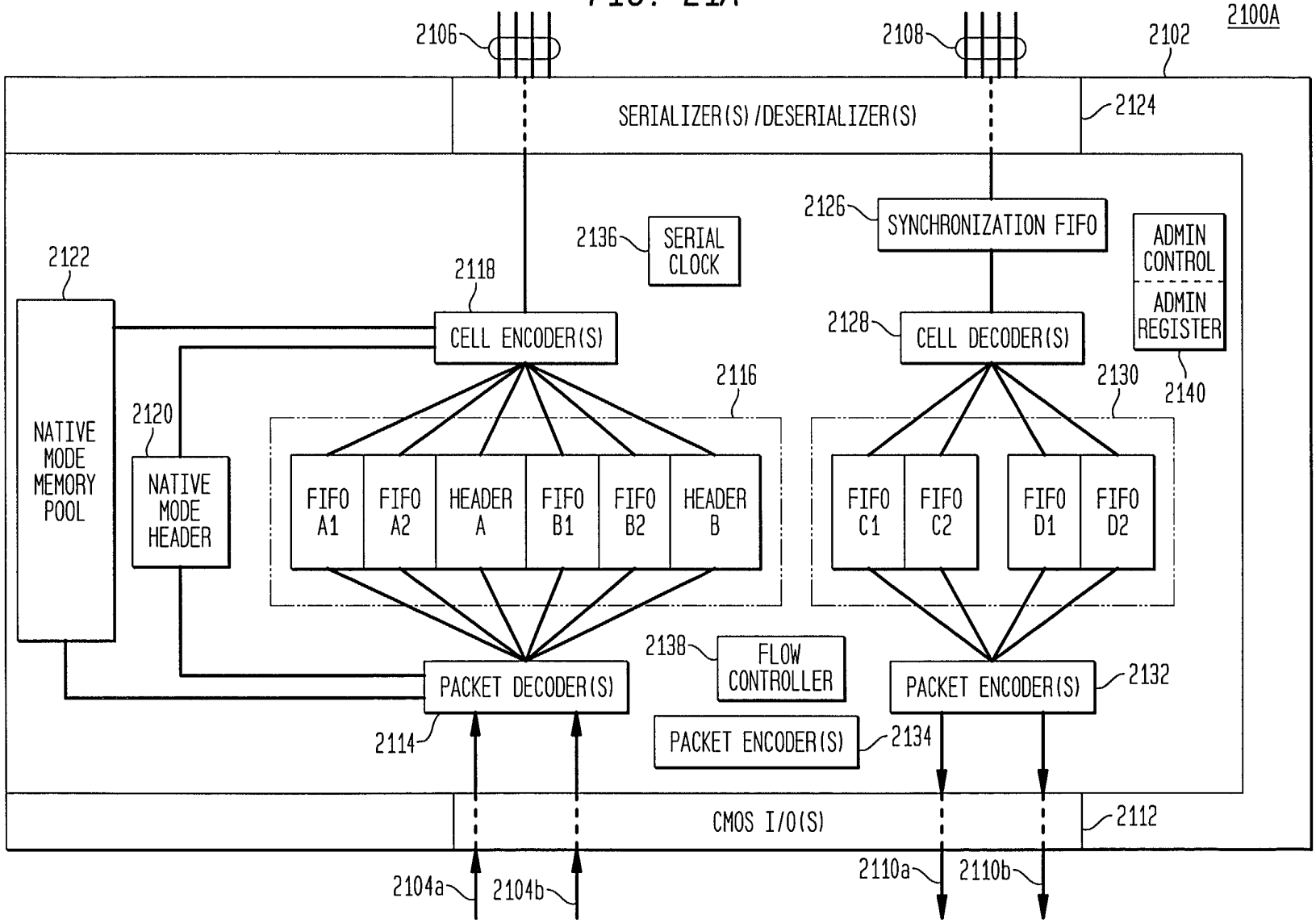

2100A
2102
2106
2108
SERIALIZER(S)/DESERIALIZER(S)
2124
2126
SYNCHRONIZATION FIFO
ADMIN CONTROL
ADMIN REGISTER
2140
2136
SERIAL CLOCK
2122
2118
CELL ENCODER(S)
2128
CELL DECODER(S)
2116
2130
NATIVE MODE MEMORY POOL
2120
NATIVE MODE HEADER
FIFO A1
FIFO A2
HEADER A
FIFO B1
FIFO B2
HEADER B
FIFO C1
FIFO C2
FIFO D1
FIFO D2
2138
FLOW CONTROLLER
PACKET DECODER(S)
PACKET ENCODER(S)
2132
2114
PACKET ENCODER(S)
2134
CMOS I/O(S)
2112
2104a
2104b
2110a
2110b

FIG. 21B

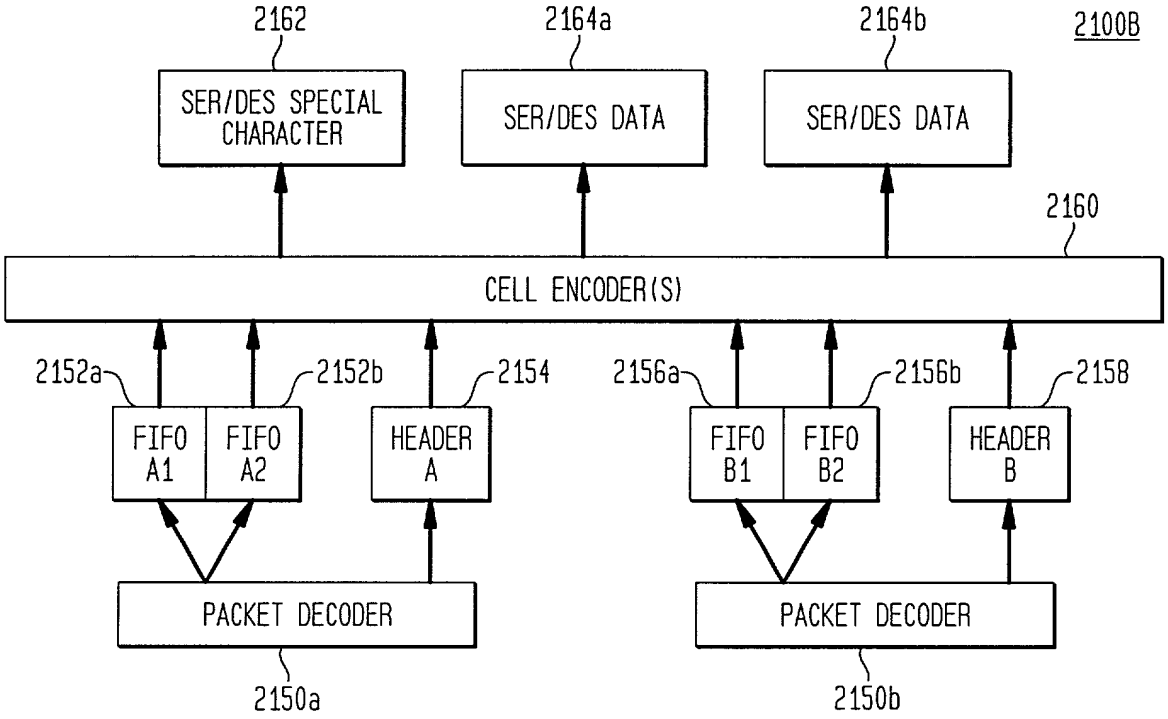


FIG. 21C

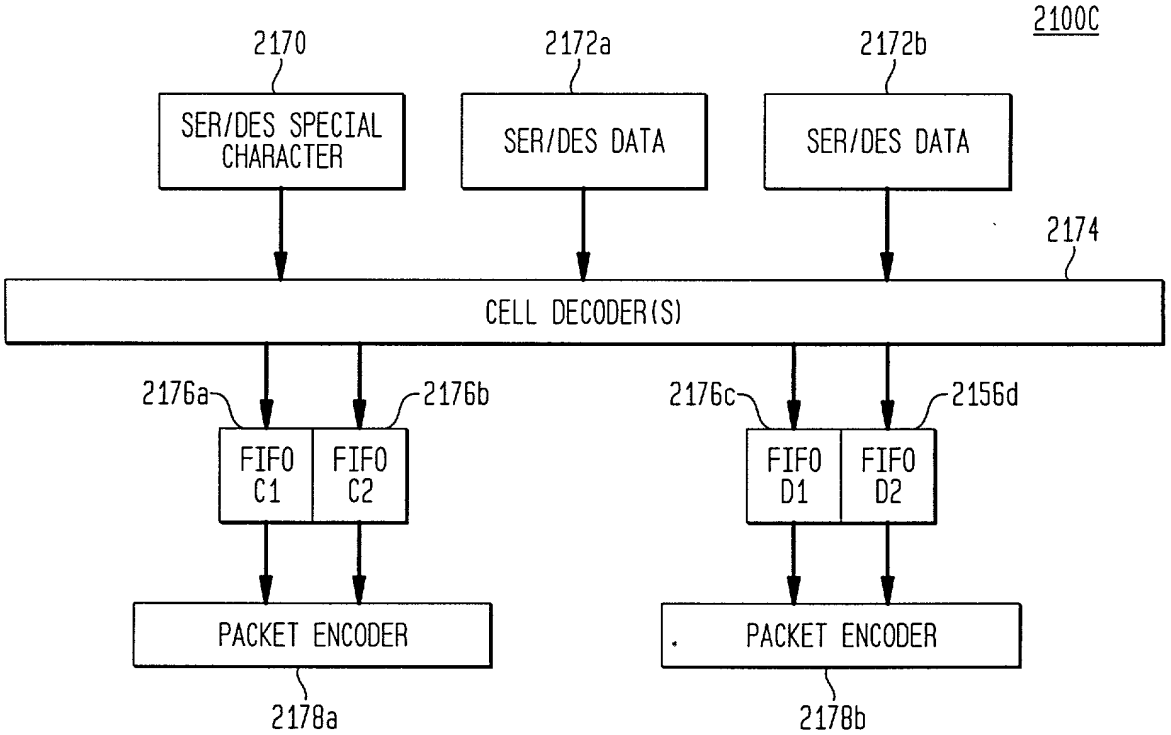


FIG. 21D

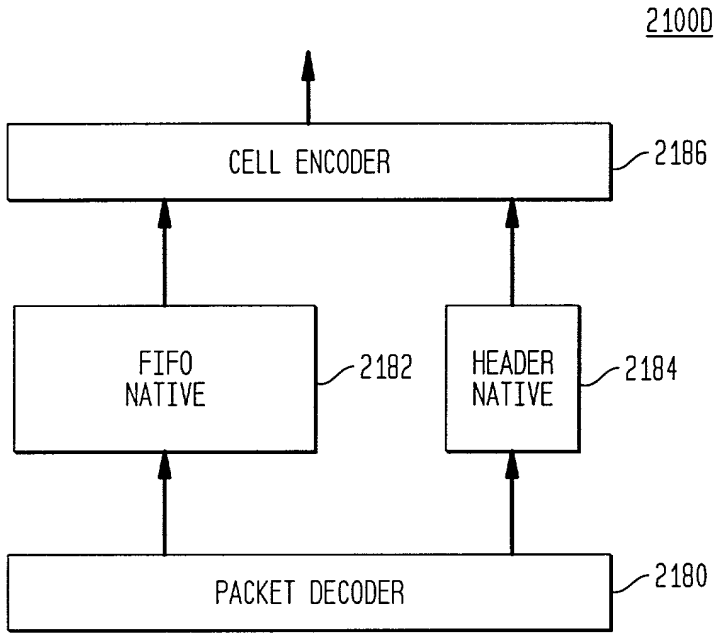

2100D
CELL ENCODER
2186
FIFO NATIVE
2182
HEADER NATIVE
2184
PACKET DECODER
2180

FIG. 21E

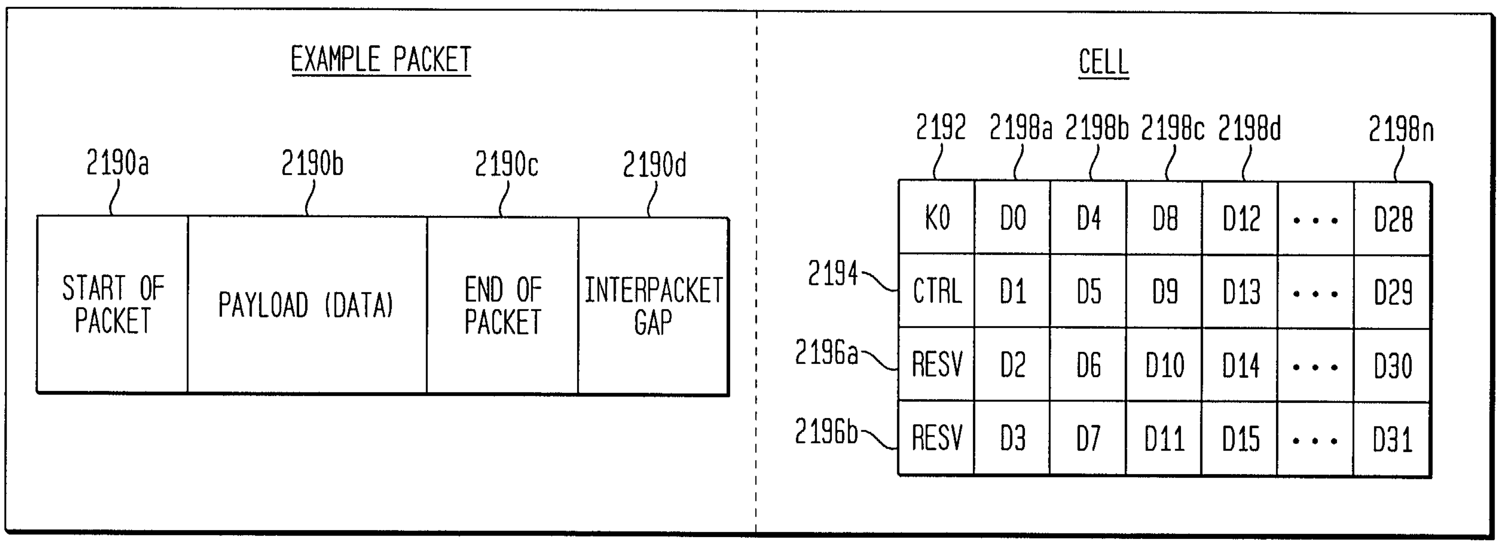

EXAMPLE PACKET
2190a
2190b
2190c
2190d
START OF PACKET
PAYLOAD (DATA)
END OF PACKET
INTERPACKET GAP
CELL
2192
2198a
2198b
2198c
2198d
2198n
2194
2196a
2196b
K0
D0
D4
D8
D12
• • •
D28
CTRL
D1
D5
D9
D13
• • •
D29
RESV
D2
D6
D10
D14
• • •
D30
RESV
D3
D7
D11
D15
• • •
D31

FIG. 22

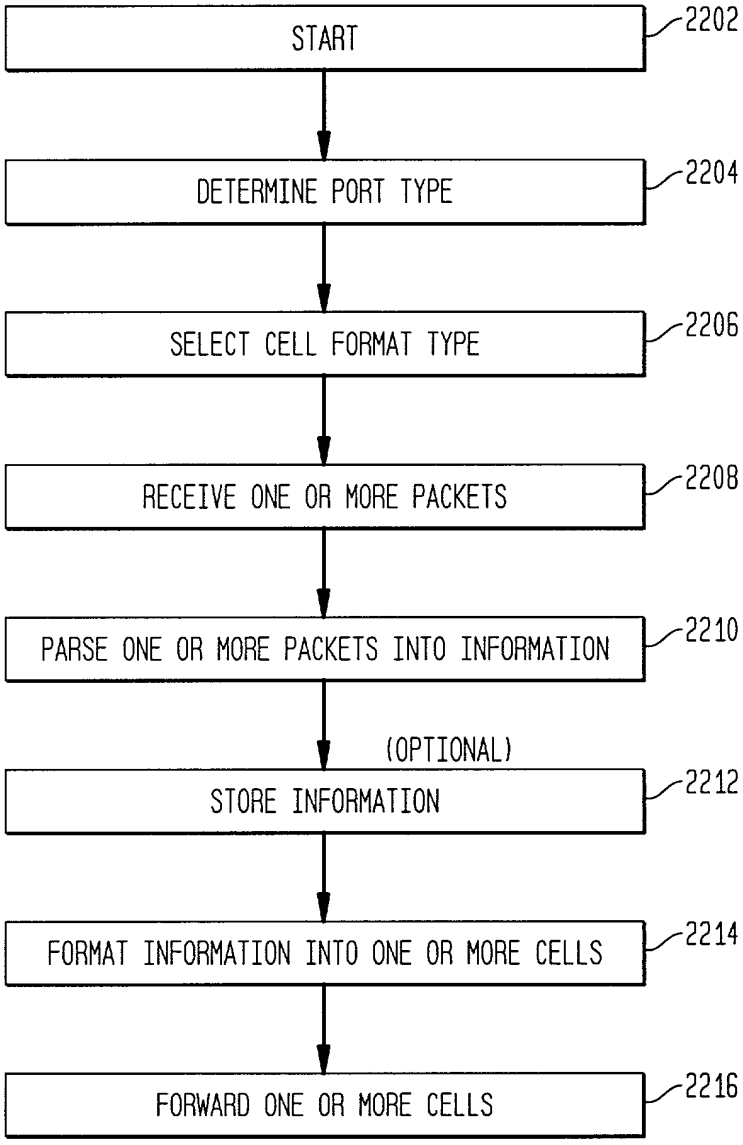

START
2202
DETERMINE PORT TYPE
2204
SELECT CELL FORMAT TYPE
2206
RECEIVE ONE OR MORE PACKETS
2208
PARSE ONE OR MORE PACKETS INTO INFORMATION
2210
(OPTIONAL)
STORE INFORMATION
2212
FORMAT INFORMATION INTO ONE OR MORE CELLS
2214
FORWARD ONE OR MORE CELLS
2216

FIG. 23A

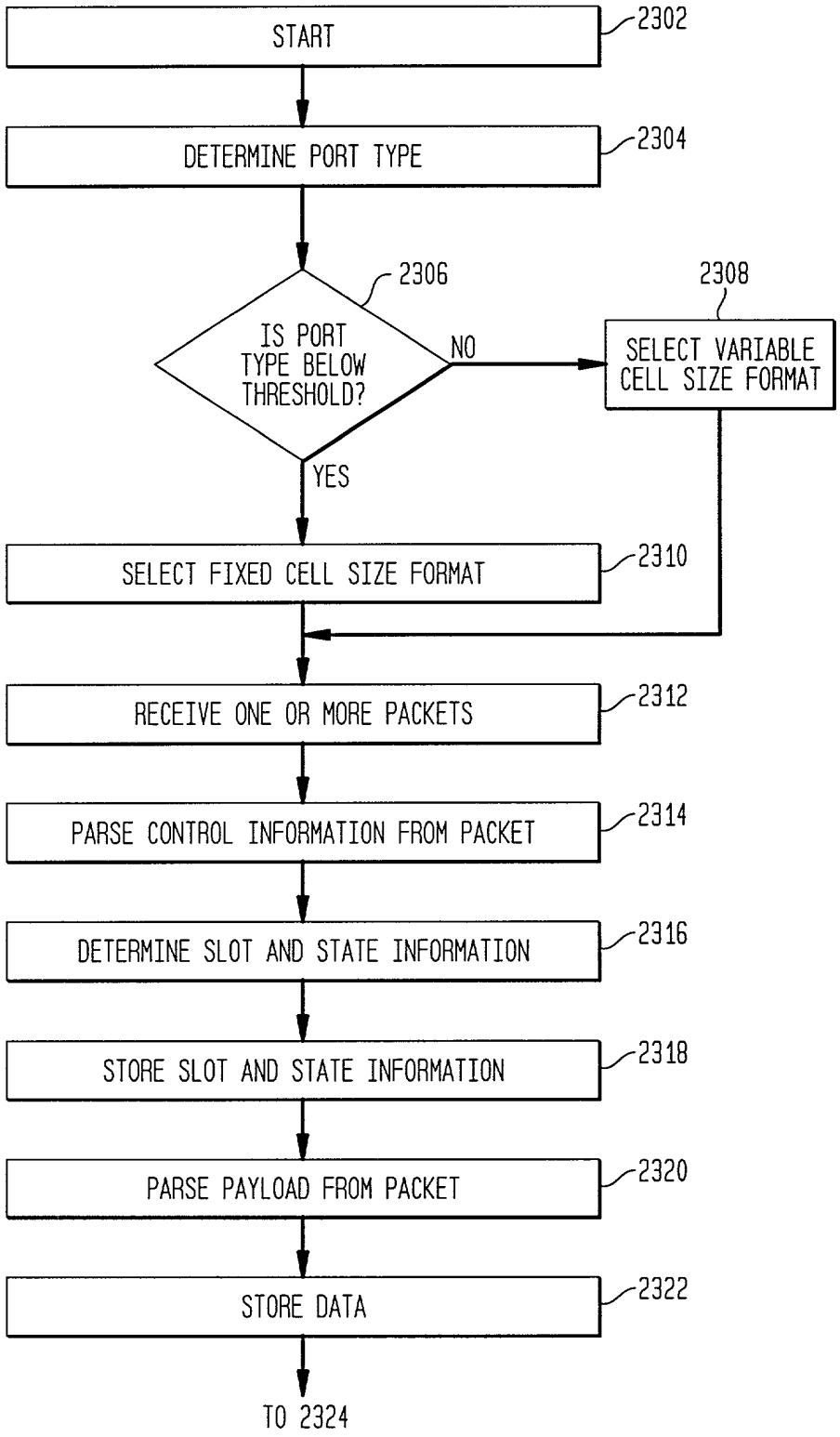

START
2302
DETERMINE PORT TYPE
2304
IS PORT TYPE BELOW THRESHOLD?
2306
NO
SELECT VARIABLE CELL SIZE FORMAT
2308
YES
SELECT FIXED CELL SIZE FORMAT
2310
RECEIVE ONE OR MORE PACKETS
2312
PARSE CONTROL INFORMATION FROM PACKET
2314
DETERMINE SLOT AND STATE INFORMATION
2316
STORE SLOT AND STATE INFORMATION
2318
PARSE PAYLOAD FROM PACKET
2320
STORE DATA
2322
TO 2324

FIG. 23B

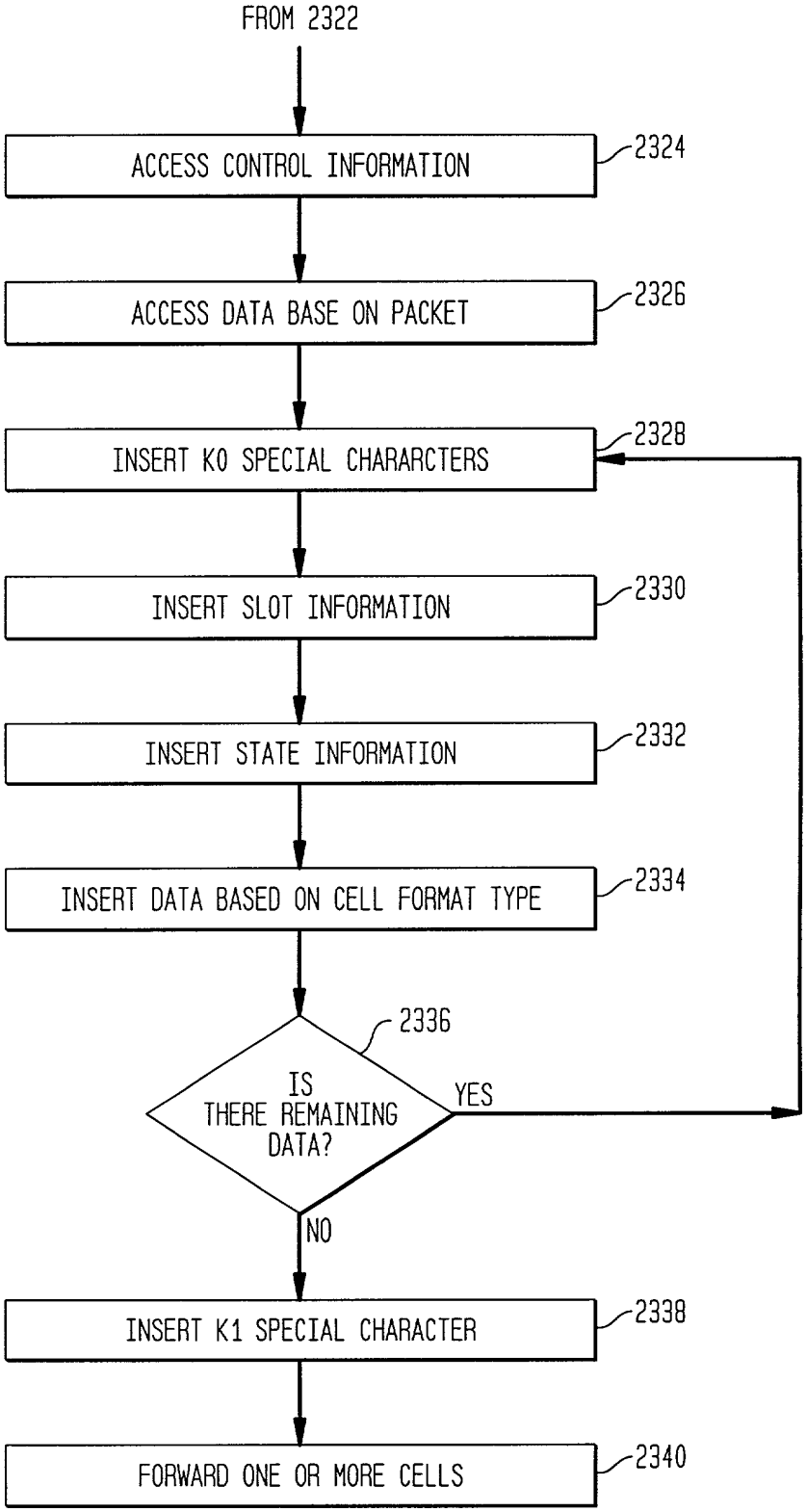

FROM 2322
ACCESS CONTROL INFORMATION
2324
ACCESS DATA BASE ON PACKET
2326
INSERT K0 SPECIAL CHARARCTERS
2328
INSERT SLOT INFORMATION
2330
INSERT STATE INFORMATION
2332
INSERT DATA BASED ON CELL FORMAT TYPE
2334
2336
IS THERE REMAINING DATA?
YES
NO
INSERT K1 SPECIAL CHARACTER
2338
FORWARD ONE OR MORE CELLS
2340

FIG. 24

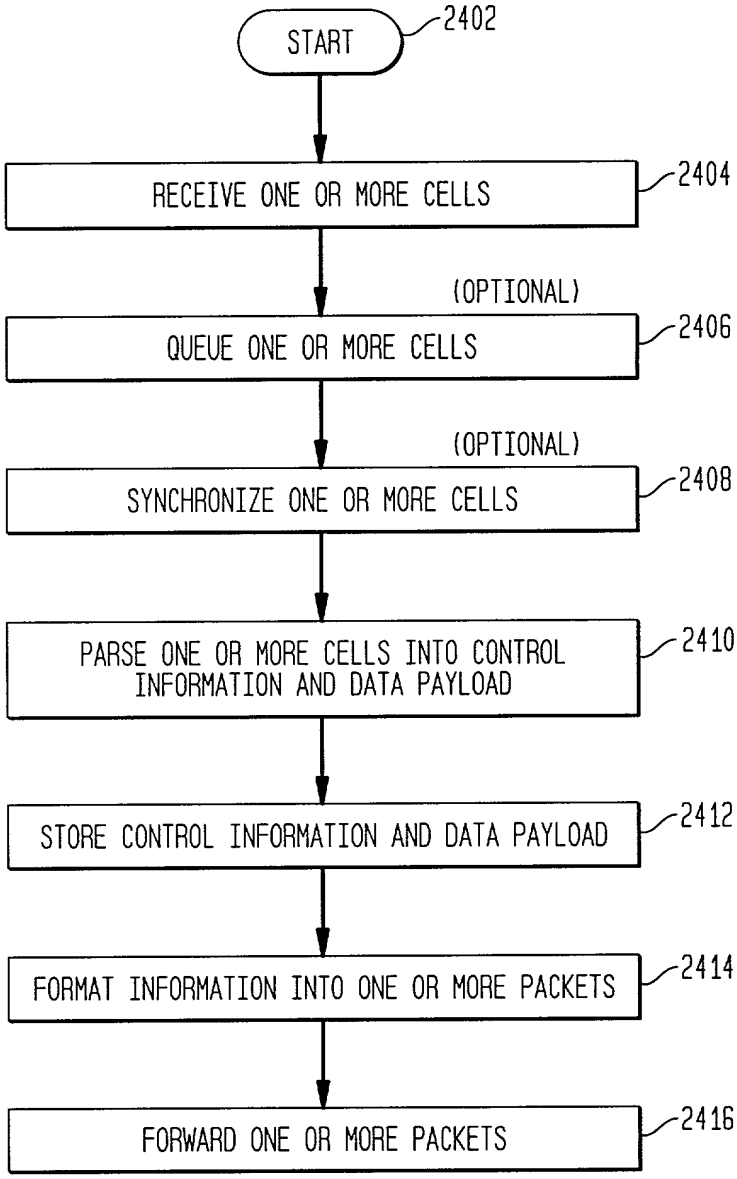

START
2402
RECEIVE ONE OR MORE CELLS
2404
(OPTIONAL)
QUEUE ONE OR MORE CELLS
2406
(OPTIONAL)
SYNCHRONIZE ONE OR MORE CELLS
2408
PARSE ONE OR MORE CELLS INTO CONTROL INFORMATION AND DATA PAYLOAD
2410
STORE CONTROL INFORMATION AND DATA PAYLOAD
2412
FORMAT INFORMATION INTO ONE OR MORE PACKETS
2414
FORWARD ONE OR MORE PACKETS
2416

FIG. 25A

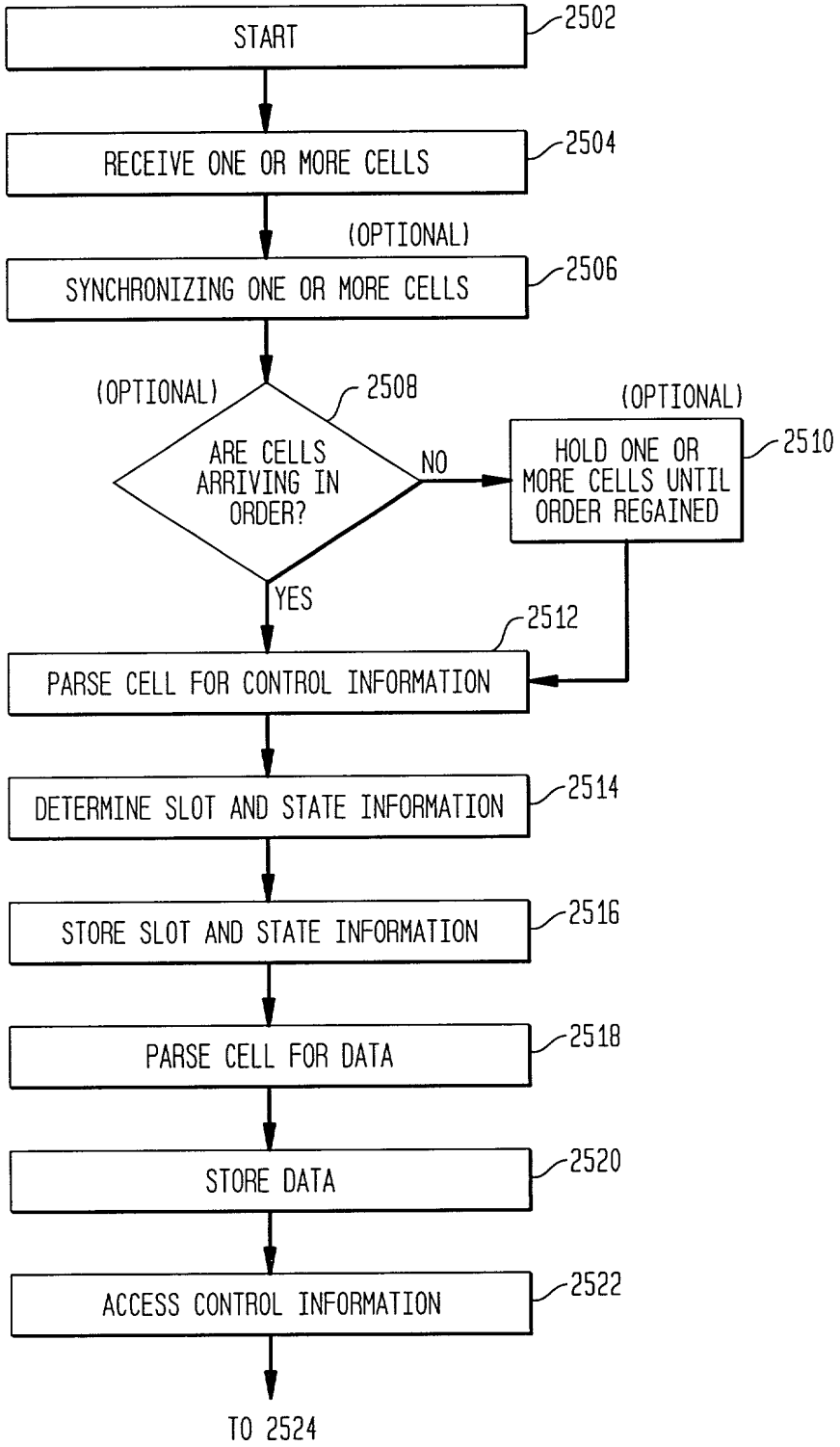

START
2502
RECEIVE ONE OR MORE CELLS
2504
(OPTIONAL)
SYNCHRONIZING ONE OR MORE CELLS
2506
(OPTIONAL)
2508
ARE CELLS ARRIVING IN ORDER?
NO
(OPTIONAL)
HOLD ONE OR MORE CELLS UNTIL ORDER REGAINED
2510
YES
2512
PARSE CELL FOR CONTROL INFORMATION
DETERMINE SLOT AND STATE INFORMATION
2514
STORE SLOT AND STATE INFORMATION
2516
PARSE CELL FOR DATA
2518
STORE DATA
2520
ACCESS CONTROL INFORMATION
2522
TO 2524

FIG. 25B

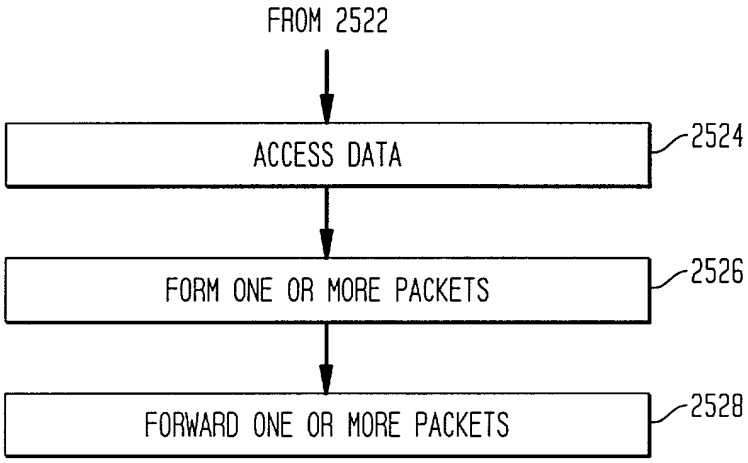


FIG. 26

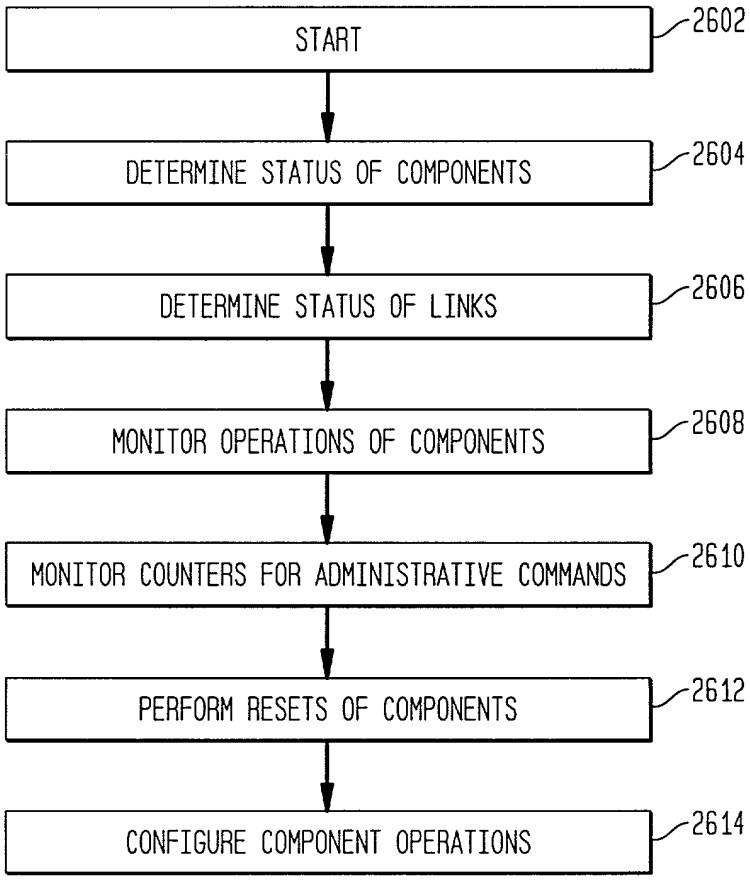


FIG. 27A

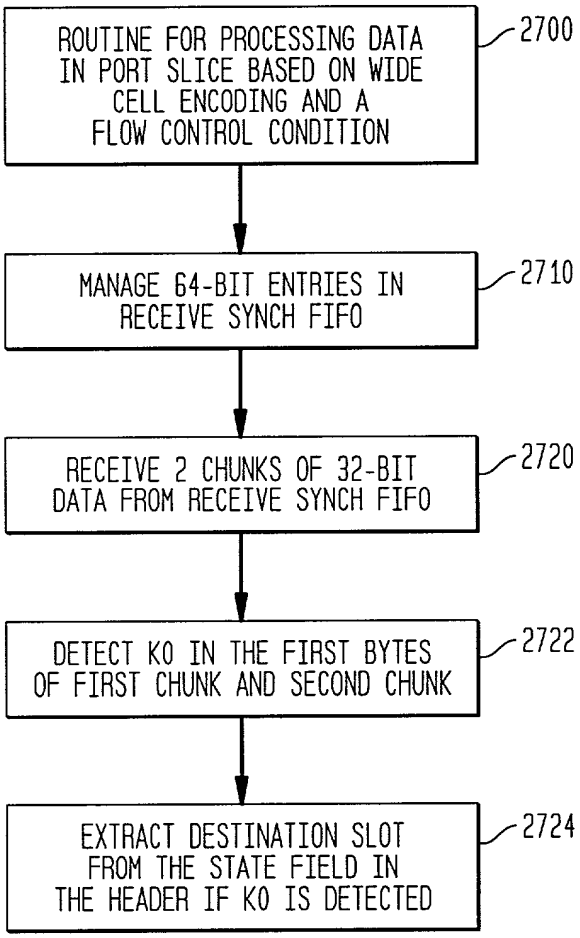

ROUTINE FOR PROCESSING DATA IN PORT SLICE BASED ON WIDE CELL ENCODING AND A FLOW CONTROL CONDITION
2700
MANAGE 64-BIT ENTRIES IN RECEIVE SYNCH FIFO
2710
RECEIVE 2 CHUNKS OF 32-BIT DATA FROM RECEIVE SYNCH FIFO
2720
DETECT K0 IN THE FIRST BYTES OF FIRST CHUNK AND SECOND CHUNK
2722
EXTRACT DESTINATION SLOT FROM THE STATE FIELD IN THE HEADER IF K0 IS DETECTED
2724

FIG. 27B

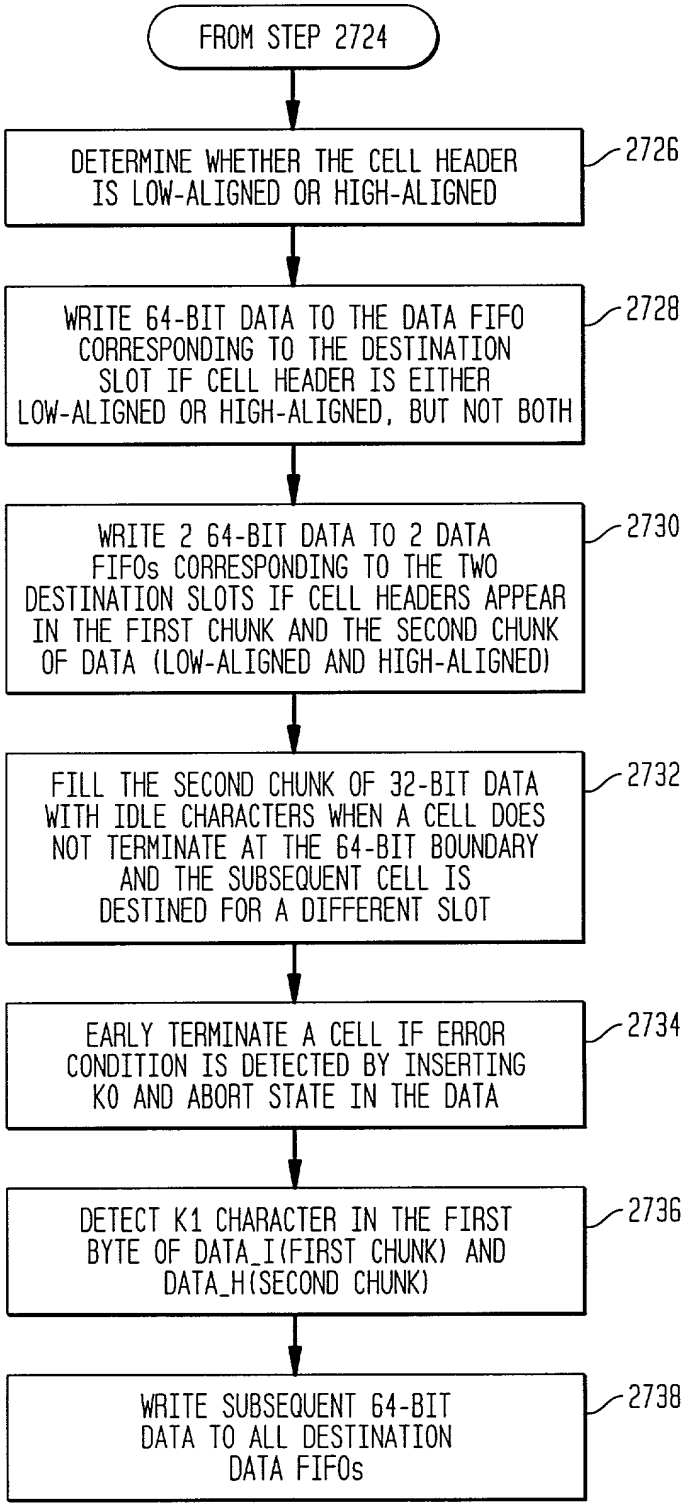

FROM STEP 2724
DETERMINE WHETHER THE CELL HEADER IS LOW-ALIGNED OR HIGH-ALIGNED
2726
WRITE 64-BIT DATA TO THE DATA FIFO CORRESPONDING TO THE DESTINATION SLOT IF CELL HEADER IS EITHER LOW-ALIGNED OR HIGH-ALIGNED, BUT NOT BOTH
2728
WRITE 2 64-BIT DATA TO 2 DATA FIFOs CORRESPONDING TO THE TWO DESTINATION SLOTS IF CELL HEADERS APPEAR IN THE FIRST CHUNK AND THE SECOND CHUNK OF DATA (LOW-ALIGNED AND HIGH-ALIGNED)
2730
FILL THE SECOND CHUNK OF 32-BIT DATA WITH IDLE CHARACTERS WHEN A CELL DOES NOT TERMINATE AT THE 64-BIT BOUNDARY AND THE SUBSEQUENT CELL IS DESTINED FOR A DIFFERENT SLOT
2732
EARLY TERMINATE A CELL IF ERROR CONDITION IS DETECTED BY INSERTING K0 AND ABORT STATE IN THE DATA
2734
DETECT K1 CHARACTER IN THE FIRST BYTE OF DATA_I(FIRST CHUNK) AND DATA_H(SECOND CHUNK)
2736
WRITE SUBSEQUENT 64-BIT DATA TO ALL DESTINATION DATA FIFOs
2738

FIG. 27C

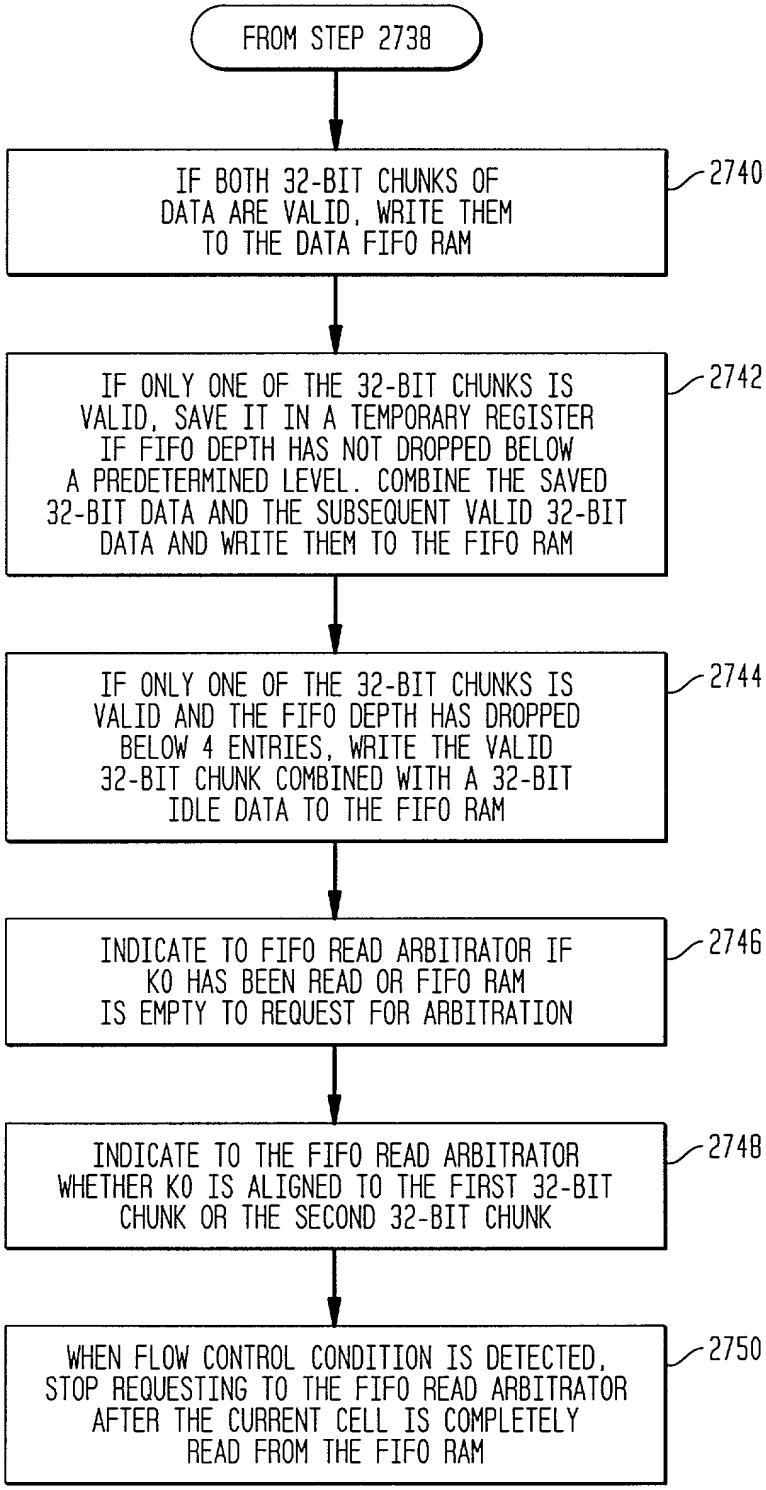

FROM STEP 2738
IF BOTH 32-BIT CHUNKS OF DATA ARE VALID, WRITE THEM TO THE DATA FIFO RAM
2740
IF ONLY ONE OF THE 32-BIT CHUNKS IS VALID, SAVE IT IN A TEMPORARY REGISTER IF FIFO DEPTH HAS NOT DROPPED BELOW A PREDETERMINED LEVEL. COMBINE THE SAVED 32-BIT DATA AND THE SUBSEQUENT VALID 32-BIT DATA AND WRITE THEM TO THE FIFO RAM
2742
IF ONLY ONE OF THE 32-BIT CHUNKS IS VALID AND THE FIFO DEPTH HAS DROPPED BELOW 4 ENTRIES, WRITE THE VALID 32-BIT CHUNK COMBINED WITH A 32-BIT IDLE DATA TO THE FIFO RAM
2744
INDICATE TO FIFO READ ARBITRATOR IF K0 HAS BEEN READ OR FIFO RAM IS EMPTY TO REQUEST FOR ARBITRATION
2746
INDICATE TO THE FIFO READ ARBITRATOR WHETHER K0 IS ALIGNED TO THE FIRST 32-BIT CHUNK OR THE SECOND 32-BIT CHUNK
2748
WHEN FLOW CONTROL CONDITION IS DETECTED, STOP REQUESTING TO THE FIFO READ ARBITRATOR AFTER THE CURRENT CELL IS COMPLETELY READ FROM THE FIFO RAM
2750

FIG. 27D

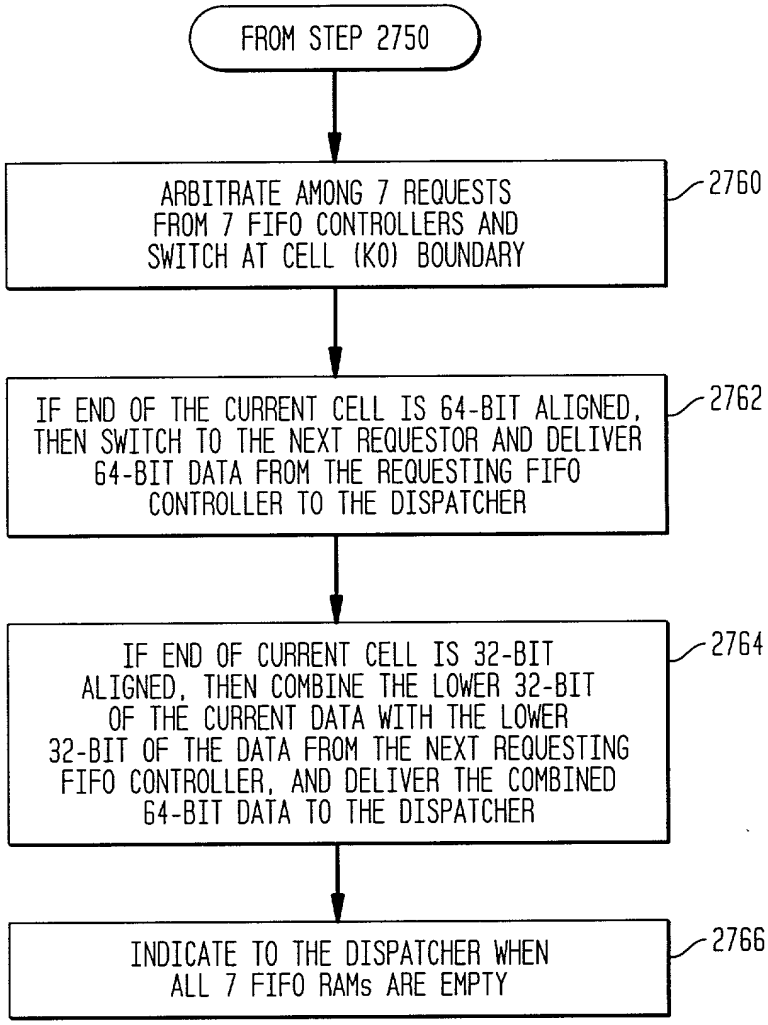
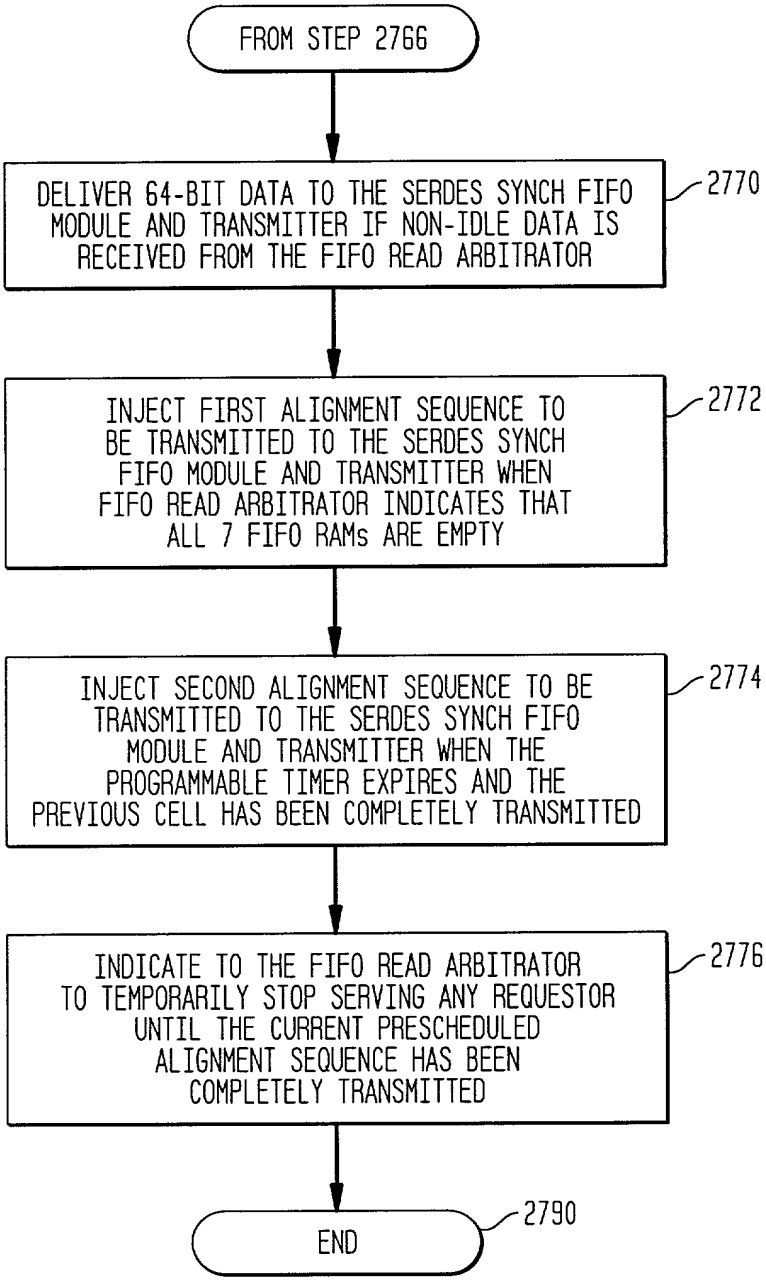

FROM STEP 2750
ARBITRATE AMONG 7 REQUESTS FROM 7 FIFO CONTROLLERS AND SWITCH AT CELL (KO) BOUNDARY
2760
IF END OF THE CURRENT CELL IS 64-BIT ALIGNED, THEN SWITCH TO THE NEXT REQUESTOR AND DELIVER 64-BIT DATA FROM THE REQUESTING FIFO CONTROLLER TO THE DISPATCHER
2762
IF END OF CURRENT CELL IS 32-BIT ALIGNED, THEN COMBINE THE LOWER 32-BIT OF THE CURRENT DATA WITH THE LOWER 32-BIT OF THE DATA FROM THE NEXT REQUESTING FIFO CONTROLLER, AND DELIVER THE COMBINED 64-BIT DATA TO THE DISPATCHER
2764
INDICATE TO THE DISPATCHER WHEN ALL 7 FIFO RAMs ARE EMPTY
2766

FIG. 27E


FROM STEP 2766
DELIVER 64-BIT DATA TO THE SERDES SYNCH FIFO MODULE AND TRANSMITTER IF NON-IDLE DATA IS RECEIVED FROM THE FIFO READ ARBITRATOR
2770
INJECT FIRST ALIGNMENT SEQUENCE TO BE TRANSMITTED TO THE SERDES SYNCH FIFO MODULE AND TRANSMITTER WHEN FIFO READ ARBITRATOR INDICATES THAT ALL 7 FIFO RAMs ARE EMPTY
2772
INJECT SECOND ALIGNMENT SEQUENCE TO BE TRANSMITTED TO THE SERDES SYNCH FIFO MODULE AND TRANSMITTER WHEN THE PROGRAMMABLE TIMER EXPIRES AND THE PREVIOUS CELL HAS BEEN COMPLETELY TRANSMITTED
2774
INDICATE TO THE FIFO READ ARBITRATOR TO TEMPORARILY STOP SERVING ANY REQUESTOR UNTIL THE CURRENT PRESCHEDULED ALIGNMENT SEQUENCE HAS BEEN COMPLETELY TRANSMITTED
2776
END
2790